



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA  
Ministry of Higher Education and Scientific Research  
MOHAMED KHIDER UNIVERSITY – BISKRA  
Faculty of Exacts Sciences, Nature and Life Sciences  
**Department of Computer Science**

Order N° : AI25/M2/2022

## Dissertation

Presented to obtain the academic master's degree in

# Computer Science

Option : Artificielle Intelligence (AI)

---

# Deep learning for Abdominal organs segmentation

---

By:

**Dounia Ferhati**

Defended on Jun 2022, in front of the jury composed of :

NAME FIRST NAME	Grade	President
NAWAL MOUAKI BENANI	Dr	Supervisor
NAME FIRST NAME	Grade	Examiner

Academic year: 2021/2022

# Acknowledgments

Firstly, i will thank, God, for guiding me through this path, giving me such wonderful family, friends, and the mentors to complete this dissertation.

I would like to express my sincere gratitude to my supervisor Dr.Nawal Mouaki Benani and Dr.IMAN YOUKANA for their continuous support of my dissertation, for their patience, motivation, and their continuous positive energy from the very first day. their guidance and observations always helped me throughout the research.

My cordial thanks extend to my dissertation committee members who accepted to evaluate this modest work.

My deepest gratitude goes to my family members who always encouraged me to do my best. Their assistance, along with their continuous motivation kept me going. I am forever thankful to my parents for being such understanding and supportive figures in my life. I can never thank them enough for their endless support and encouragement especially in difficult times.

# Abstract

Abdominal organ segmentation, which is, the delineation of organ areas in the abdomen, plays an important role in the process of radiological evaluation. It try to automatically segment the abdominal organs which will aid radiologists who are required to view thousands of images daily. This study outlines the current state-of-the-art of automated methods used to segment abdominal organ regions from computed tomography (CT) and also magnetic resonance imaging (MRI). This task of segmentation is a challenge because the abdomen CT and MRIs are severely affected by intensity non uniformity, complex organ, and low contrast during acquisition.

In our work, we build a deep-learning based methods (the U-net, Vgg16-Unet and Vgg19-Unet networks) in order to automatically segment the abdominal organs (liver, the right kidney,the left kidney and the spleen) in both CT and MRI modalities imaging. In the 3D MRI multi sequences (T1-DUALint, T1-DUALout and T2-SPIR), the segmentation will be into the four organs. Whereas, for the 3D CT images, the segmentation is just into liver organ label. For this study, we used the dataset provided in the CHOAS Challenge [1] for evaluated our models.

The obtained results for each modality across different subjects are satisfying comparing to some existing deep learning-based approaches.

**Key-words:** abdominal organs segmentation, Deep Learning, U-net, Vgg16-Unet, Vgg19-Unet, CHOAS, Computed tomography (CT), Resonance imaging (MRI) .

# Résumé

La segmentation des organes abdominaux, c'est-à-dire la délimitation des zones des organes dans l'abdomen, joue un rôle important dans le processus d'évaluation radiologique. Il essaie de segmenter automatiquement les organes abdominaux, ce qui aidera les radiologues qui doivent voir des milliers d'images chaque jour. Cette étude décrit l'état actuel des techniques automatisées utilisées pour segmenter les régions des organes abdominaux à partir de la tomodensitométrie (TDM) et de l'imagerie par résonance magnétique (IRM). Cette tâche de segmentation est un défi parce que la TDM abdominale et les IRM sont gravement affectées par la non-uniformité de l'intensité, l'organe complexe et le faible contraste au cours de l'acquisition.

Dans notre travail, nous construisons des méthodes basées sur l'apprentissage profond (les réseaux U-net, Vgg16-Unet et Vgg19-Unet) afin de segmenter automatiquement les organes abdominaux (foie, rein droit, rein gauche et rate) dans les modalités de TDM et IRM. Dans les séquences multiples d'IRM 3D (T1-DUALint, T1-DUALout et T2-SPIR), la segmentation se fera dans les quatre organes. Alors que, pour les images 3D CT, la segmentation se limite à l'étiquette de l'organe hépatique. Pour cette étude, nous avons utilisé l'ensemble de données fourni dans le Défi CHOAS [1] pour évaluer nos modèles.

Les résultats obtenus pour chaque modalité sur différents patients sont satisfaisants par rapport à certaines approches existantes basées sur le deep learning.

**Mots-clés** : segmentation des organes abdominaux, Deep Learning, les réseaux U-net, Vgg16-Unet et Vgg19-Unet, base d'images CHAOS, Tomodensitométrie (TDM), Résonance magnétique (IRM).

# Contents

<b>List of Figures</b>	<b>IV</b>
<b>General introduction</b>	<b>1</b>
<b>1 Abdominal organs segmentation in computed tomography (CT) and magnetic resonance (MR) image</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 Medical Imaging . . . . .	4
1.3 Imaging Modalities for Abdomen . . . . .	4
1.3.1 Computed Tomography Imaging (CT) . . . . .	4
1.3.2 Magnetic Resonance Imaging (MRI) . . . . .	6
1.3.3 MRI imaging sequences . . . . .	7
1.3.3.1 T1-weighted Image . . . . .	7
1.3.3.2 T2-Weighted Image . . . . .	7
1.4 Human Abdominal Anatomy . . . . .	8
1.4.1 Liver . . . . .	8
1.4.2 kidneys . . . . .	10
1.4.3 Spleen . . . . .	10
1.5 Abdominal CT and MRI images segmentation . . . . .	10
1.5.1 Abdominal CT and MRI images segmentation approaches . . . . .	11
1.5.1.1 Semi-automatic methods . . . . .	12
1.5.1.2 Automatic Methods . . . . .	13
1.6 Conclusion . . . . .	15
<b>2 Deep learning</b>	<b>16</b>
2.1 Introduction . . . . .	17
2.2 Machine learning (ML) . . . . .	17
2.2.1 Types of Machine learning Algorithms . . . . .	17
2.2.1.1 Supervised learning . . . . .	17
2.2.1.2 Unsupervised learning . . . . .	18
2.2.1.3 Semi-supervised Learning . . . . .	19

2.2.1.4	Reinforcement learning . . . . .	19
2.2.1.5	Deep learning . . . . .	19
2.2.1.6	Transfer Learning . . . . .	21
2.2.2	Artificial Neural Networks . . . . .	21
2.3	Convolution Neural Networks (CNNs) . . . . .	22
2.3.1	The Basics of Convolutional Neural Network . . . . .	23
2.3.1.1	Convolutional Layer . . . . .	23
2.3.1.2	Pooling Layer . . . . .	24
2.3.1.3	Fully Connected Layer . . . . .	24
2.3.2	Activation Functions . . . . .	24
2.3.2.1	Rectified Linear Unit (ReLU) . . . . .	24
2.3.2.2	SIGMOID . . . . .	25
2.4	U-Net network . . . . .	25
2.5	Deep Models Matter . . . . .	27
2.5.1	Visual Geometry Group (VggNet) . . . . .	27
2.5.2	Residual Network (ResNet) . . . . .	27
2.5.3	Densely Connected Network (DenseNet) . . . . .	27
2.6	Advanced Training Techniques . . . . .	27
2.6.1	Data Augmentation . . . . .	28
2.6.2	Regularization . . . . .	28
2.7	Evaluation Metrics . . . . .	28
2.8	Related Works of abdominal multi-organs segmentation . . . . .	30
2.9	Conclusion . . . . .	31
<b>3</b>	<b>Design, Implementation and experimental results</b>	<b>32</b>
3.1	Introduction . . . . .	33
3.2	Design of Abdominal Multi-organ Segmentation using Deep learning . . . . .	33
3.3	Global design . . . . .	33
3.4	Detailed design . . . . .	34
3.4.1	Dataset preparation . . . . .	35
3.4.1.1	Collecting dataset . . . . .	35
3.4.1.2	Loading and Pre-processing . . . . .	35
3.4.1.3	Data splitting . . . . .	35
3.4.2	Training phase of our deep model . . . . .	36
3.4.2.1	Prediction phase . . . . .	37
3.5	Implementation . . . . .	38
3.5.1	Environments and developing tools . . . . .	38
3.5.1.1	Hardware configuration . . . . .	38
3.5.1.2	Development environment . . . . .	38

3.5.1.3	Packages and APIs . . . . .	39
3.5.2	Details informations of used Image dataset . . . . .	40
3.5.3	Implementation details . . . . .	41
3.5.3.1	Data preparation and loading . . . . .	41
3.5.3.2	Creating of our Deep Learning Models . . . . .	46
3.5.3.3	Compiling Model . . . . .	46
3.5.3.4	Data augmentation function . . . . .	47
3.5.3.5	Training Models . . . . .	47
3.5.3.6	Prediction phase . . . . .	48
3.5.3.7	Visualising Results . . . . .	48
3.6	Experimental Results . . . . .	48
3.6.1	The obtained segmentation results of the liver organ . . . . .	49
3.6.1.1	3D MRI imaging modality Database . . . . .	49
3.6.1.2	CT Database . . . . .	52
3.6.2	The obtained segmentation results of spleen organ in MRI database . . . . .	53
3.6.3	The obtained segmentation results of right kidney organ in MRI database . . . . .	57
3.6.4	The obtained segmentation results of Left Kidney in MRI database . . . . .	60
3.6.5	Results Comparison . . . . .	64
3.6.6	Discussion . . . . .	65
3.7	Conclusion . . . . .	65
	<b>General Conclusions</b>	<b>66</b>
	<b>Bibliography</b>	<b>67</b>

# List of Figures

1.1	Exemple of CT Machine. . . . .	5
1.2	The principle of computed tomography. . . . .	6
1.3	abdominal MR images Planes: Axial, Sagittal, Coronal. . . . .	7
1.4	Illustration of examples of abdominal MR images sequences. . . . .	8
1.5	Illustration of human Abdominal organs. . . . .	9
1.6	Anatomy of Liver:(a) parts of liver ,(b) location of liver. . . . .	9
1.7	Anatomy of Kidney: (a) parts of kidney ,(b) location of kidney. . . . .	10
1.8	Anatomy of Spleen. . . . .	11
1.9	Abdominal segmentation using thresholding method: (a) original CT image, (b) Results of segmentation. . . . .	12
1.10	Abdominal segmentation using region based methods:(a) original CT image ,(b) segmentation results. . . . .	13
1.11	Abdominal CT image:(a) original CT image ,(b) CT image using edge meth- ods . . . . .	14
2.1	The processes of supervised Machine Learning. . . . .	18
2.2	Examples of supervised and unsupervised learning models. . . . .	19
2.3	Reinforcement learning. . . . .	20
2.4	Categories of Deep learning methods. . . . .	21
2.5	Diagram of the artificial neuron[35]. . . . .	22
2.6	Example of convolutional layer slides [34]. . . . .	23
2.7	Illustration of different types of pooling operation [38]. . . . .	24
2.8	ReLU Activation Function Plot [40]. . . . .	25
2.9	SIGMOID Activation Function Plot [40]. . . . .	26
2.10	U-Net Architecture [43]. . . . .	26
3.1	Global design of our abdominal segmentation system. . . . .	34
3.2	Detailed design of abdominal segmentation system. . . . .	34
3.3	Illustration of training phase of our deep model. . . . .	37
3.4	Illustration of Prediction phase. . . . .	37
3.5	Examples of images from CT database. . . . .	41
3.6	Examples of MR images from database. . . . .	41

3.7	Illustration of used packages and modules. . . . .	42
3.8	Illustration of used function of Reading the MRI data. . . . .	42
3.9	Pre-processing data function. . . . .	43
3.10	Illustration of examples of 2D slices with organs contours and their correspondents masks. . . . .	44
3.11	Function used to save data on Numpy files. . . . .	45
3.12	Normalization function. . . . .	45
3.13	CT Loading and pre-processing functions. . . . .	45
3.14	Function of Splitting the data . . . . .	46
3.15	Illustration of the data augmentation function. . . . .	47
3.16	Illustration of the fitting model function. . . . .	47
3.17	Illustration of Visualising Results function. . . . .	48
3.18	Illustration of dice and loss coefficients of Unet Model for Liver segmentation in MRI modality. . . . .	50
3.19	Illustration of dice and loss coefficients of Vgg16-Unet Model for Liver segmentation in MRI modality . . . . .	50
3.20	Illustration of dice and loss coefficients of Vgg19-Unet Model for Liver segmentation in MRI modality. . . . .	51
3.21	Illustration of the predicted results of Unet, Vgg16-Unet models for Liver segmentation in MRI. a) 2D axial slice, (b) Ground Truth, (c) our predicted results. . . . .	51
3.22	Illustration of the predicted results of Vgg19-Unet model for liver segmentation in MRI. a) 2D axial slice, (b) Ground Truth, (c) our predicted results. . . . .	52
3.23	Illustration of the obtained performance (dice and loss coefficients) of Unet Model for Liver organ segmentation in CT images. . . . .	52
3.24	Illustration of the predicted results of Unet models for liver organ segmentation in CT images. a) 3D CT images, (b) Ground Truth, (c) our predicted results. . . . .	53
3.25	Illustration the dice and loss coefficient of Vgg16-Unet model for Spleen organ in MRI database. . . . .	54
3.26	Illustration the dice and loss coefficient of Unet model for Spleen organ in MRI database. . . . .	54
3.27	Illustration the dice and loss coefficient of Vgg19-Unet model for spleen organ in MRI database. . . . .	55
3.28	Illustration the the predicted results of Vgg16-Unet model for spleen organ in MRI database. a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . . . .	55

3.29	Illustration the the predicted results of Unet model for Spleen organ in MRI database. a) 2D axial slices, (b) Ground Truth, (c) our predicted results. . .	56
3.30	Illustration the the predicted results of Vgg19-Unet model for Spleen organ in MRI database. a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . . . .	56
3.31	Illustration the dice and loss coefficient of Vgg16-Unet model for Right Kidney organ in MRI database . . . . .	57
3.32	Illustration the dice and loss coefficient of Unet model for Right Kidney organ.	58
3.33	Illustration the dice and loss coefficient of Vgg19-Unet model for Right Kidney organ in MRI database . . . . .	58
3.34	Illustration the the predicted results of Vgg16-Unet model for Right Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . .	59
3.35	Illustration the the predicted results of Unet model for Right Kidney organ. a) 2D axial slices, (b) Ground Truth, (c) our predicted results. . . . .	59
3.36	Illustration the the predicted results of Vgg19-Unet model for Right Kidney organ. a) 2D axial slices, (b) Ground Truth, (c) our predicted results. . . .	60
3.37	Illustration the dice and loss coefficient of Vgg16-Unet model for Left Kidney organ in MRI. . . . .	61
3.38	Illustration the dice and loss coefficient of Unet model for Left Kidney organ in MRI. . . . .	61
3.39	Illustration the dice and loss coefficient of Vgg19-Unet model for Left Kidney organ in MRI. . . . .	61
3.40	Illustration the the predicted results of Vgg16-Unet model for Left Kidney organ.a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . . .	62
3.41	Illustration the the predicted results of Unet model for Left Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . . . .	63
3.42	Illustration the the predicted results of Vgg19-Unet model for Left Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results. . .	63

# General introduction

Medical imaging is a rapidly evolving field that is deeply intertwined with many other disciplines. Recent medical imaging discoveries not only allow for better diagnosis, but also provide new treatment options for a variety of diseases such as cancer. One of the most significant processes in medical image processing is the segmentation. It consists the division of an image into various sections and helps to locate areas of interest in an image. As well as making it easier and more relevant to evaluate. The segmentation of images is the primary goal of every computer vision system and a key step in the image analysis process.

The segmentation of objects in images has undoubtedly advanced in the last twenty years in the field of medicine. These recent discoveries not only allow for better diagnosis, but also provide new treatment options for a variety of diseases.

Today, radiologists, with the assistance of anatomical observation tools such as Computed topography (CT) and Magnetic resonance (MR), enable surgeons to perform intricate and delicate surgical operations in a less intrusive manner. Computed topography (CT) and Magnetic resonance (MR) imaging are the most widely used radiography techniques in diagnosis, clinical studies and treatment planning.

Human abdominal organ segmentation with computed tomography (CT) and magnetic resonance imaging (MRI) is widely used for measuring and visualising diverse abdomen structures, highlighting lesions, image guided procedures, and surgical planning. Internal organ location and structural information are crucial in a variety of therapeutic applications, including radiation therapy (RT), imaging-guided surgery, and lesion quantification. This job is tough because non-uniform intensity, intricate organs, and low contrast during collection have a negative impact on abdominal CT and MR images.

Manual segmentation is a necessary and fundamental step in all treatment and diagnostic procedures, but not always obvious due to several artefacts and other peculiarities related to the characteristics of the object to be segmented (abdomen anatomy), making segmentation a laborious, difficult process and consumes long time. As a result, it is necessary to have automatic diagnostic tools on hand. These assistance systems provide radiologists with precise information on the characteristics of areas of interest. A variety of approaches has been presented in order to automatically segment abdominal organs such as, the edge based methods, the region based methods, the machine learning methods and

deep learning methods and also other approaches as the Atlas methods. Recently, the deep learning is considered as one of the robust tools in image segmentation. It is a field that have spurred an overwhelming amount of research that have demonstrated performance with very high accuracy.

The goal of our work is to provide a tool for the automatic segmentation of CT and MRI abdominal images based on extensive learning (deep learning). In this study, we use three deep learning models which are the UNet with two deeper versions Vgg16-Unet and Vgg19-Unet to segment organs from CT and 3D MRI volumes into four main classes that are the liver, the spleen, the right kidney and the left kidney, looking to improve the precision of the results by varying the model's parameters.

The remainder of this dissertation is organized into three chapters as follow :

— **The chapter 01: Segmentation of human abdominal organs in computed tomography (CT) and magnetic resonance (MR) image**

This chapter is devoted to the medical imaging, presenting the imaging modalities such as computed tomography (CT) and magnetic resonance (MRI) with it different properties, and the fundamentals of human abdominal anatomy with talking about the approaches proposed for abdominal segmentation.

— **The chapter 02: Deep learning**

We will present the fundamental concepts of convolution neural networks and deep learning segmentation approaches. Starting with the machine learning and its algorithms, deep neural network until we get to the deep learning and the algorithms used. Then, we provide a description of two important methods of deep learning models: which are the CNN and the U-net networks with some deep models matter. In addition, we talking about the transfer learning and data augmentation. Next, we will provide some existing works about the abdominal organ segmentation using deep learning models.

— **The chapter 03: Design, Implementation and experimental results**

This final chapter is dedicated to presenting the results obtained after setting the global and detail conception of our system and implementing the deep learning models that we used. We will also provide a comparison of the results obtained with various architectures. Furthermore, we discuss these results in the following sections.

We conclude our report with a general conclusion and some future perspectives for this research project.

# Chapter 1

## Abdominal organs segmentation in computed tomography (CT) and magnetic resonance (MR) image

### Contents

---

<b>1.1</b>	<b>Introduction</b>	<b>4</b>
<b>1.2</b>	<b>Medical Imaging</b>	<b>4</b>
<b>1.3</b>	<b>Imaging Modalities for Abdomen</b>	<b>4</b>
1.3.1	Computed Tomography Imaging (CT)	4
1.3.2	Magnetic Resonance Imaging (MRI)	6
1.3.3	MRI imaging sequences	7
<b>1.4</b>	<b>Human Abdominal Anatomy</b>	<b>8</b>
1.4.1	Liver	8
1.4.2	kidneys	10
1.4.3	Spleen	10
<b>1.5</b>	<b>Abdominal CT and MRI images segmentation</b>	<b>10</b>
1.5.1	Abdominal CT and MRI images segmentation approaches	11
<b>1.6</b>	<b>Conclusion</b>	<b>15</b>

---

## **1.1 Introduction**

Medical images serve an important role in supporting health care providers in accessing patients for diagnosis and treatment such as Magnetic Resonance Imaging (MRI) and Computed Tomography Imaging (CT). The latter, through many research, we find that it deals with a process called the segmentation.

The segmentation of human abdominal organs in computed tomography (CT) and magnetic resonance (MR) images used in many clinical applications and in diagnosis of abdomen diseases to measure and visualize various abdominal tissues, to delineate lesions, and to design image-guided therapies and surgeries.

This chapter begins with a brief introduction about the human abdominal organs, followed by presenting two important imaging modalities and their characteristics that are abdominal CT and MR images. Following that, we discuss several classical methodologies and strategies for segmenting abdominal organs in MR images that have been presented in the literature.

## **1.2 Medical Imaging**

Medical imaging is without a doubt one of the fields of medicine that has advanced the most in the last two decades. These recent discoveries not only allow for better diagnosis, but they also provide new treatment options for a variety of diseases such as cancer, epilepsy, Alzheimer,... A precise diagnosis of the lesion facilitates access to surgery, which is the only treatment option for certain patients. Such approaches also aid in the better understanding of the operation of some yet mysterious organs [2]. Medical image processing is a useful tool for diagnosing, planning surgeries, and avoiding illnesses and difficulties in many human organs such as the brain, heart, kidney, and liver [3].

## **1.3 Imaging Modalities for Abdomen**

In order to treat the human abdomen, we find two important imaging modalities that are the computed tomography (CT) and magnetic resonance imaging (MRI) modalities. In this section, we will define the both modalities and their advantages in the abdominal segmentation.

### **1.3.1 Computed Tomography Imaging (CT)**

Computed tomography research is currently as interesting as it was when it was first developed in the 1960s and 1970s [4]. It is a widely utilized imaging technique in clinical practice. CT creates a three-dimensional (3D) picture collection that depicts the anatomy of

a patient. CT scanners provide a series of two-dimensional (2D) axial cross section pictures that may be layered to create a three-dimensional (3D) data set. CT image reconstruction mathematics is also used in a variety of other imaging modalities [5].

A typical CT image is made up of 512 rows, each with 512 pixels, for a total of 262,144 pixels (one for each voxel). The value of the attenuation coefficient for each voxel that corresponds to these pixels must be determined throughout the image processing. The CT scanner produces a digital image, which is made up of a square matrix of elements (pixels), each of which represents a voxel (volume element) of the patient's tissue.

CT differs from conventional medical imaging technology in that it is safer, less expensive, non-invasive, and non-traumatic [3].



Figure 1.1 – Exemple of CT Machine.

These technologie have a high resolution on soft tissue in the human body and can compute quickly. The gantry, a circular, rotating frame containing an X-ray tube on one side and a detector on the other, is the most visible component of a CT scanner. The revolving frame revolves the X-ray tube and detector around the patient, creating a fan-shaped beam of X-rays. Several thousand pictures are acquired in one revolution while the scanner spins, producing in one full cross-sectional image of the body (see Figure 1.2). It is possible to construct a 3D representation and views from various angles based on this data [6].

In this technique we have many advantages, so We will cite a few key ones, such as:

- Using it in many large hospitals and medical centers throughout the world.

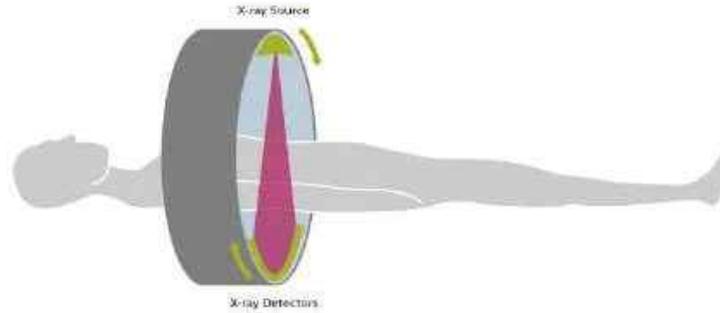


Figure 1.2 – The principle of computed tomography.

- Significantly more data is collected.
- Far easier to separate soft tissues other than bone from one another (e.g. liver, kidney).
- Adds enormously to the diagnostic information.
- Data exist in digital form so can be analyzed quantitatively.

### 1.3.2 Magnetic Resonance Imaging (MRI)

Over the last 30 years, magnetic resonance imaging (MRI) has evolved from a promising method to the major diagnostic investigation for a wide range of clinical issues. Its use, which was once confined to the neuro-axis, has expanded to include all parts of the body, and a growing body of information has led to a greater understanding of how it may be used most effectively, either alone or in combination with other techniques, to optimize diagnostic confidence. MRI is a non-invasive means of mapping the interior structure of the body as well as some aspects of function [7].

The doctors Edward Mills Purcell and Felix Bloch discovered the physical phenomenon of nuclear magnetic resonance, which is the basis for this procedure. The earliest use of magnetic resonance imaging in medicine was in the 1970s. Disease detection, diagnosis, and therapy monitoring are all common applications for MRI. It works by exciting and detecting changes in protons in the water that makes up biological tissues [8].

An MRI scanner employs strong magnets to polarize and excite hydrogen nuclei (single protons) in human tissue, resulting in a signal that can be detected and spatially recorded, resulting in pictures of the body. The MRI machine sends out a radio frequency (RF) pulse that exclusively bonds to hydrogen molecules. The pulse is sent to the precise location of

the body that has to be evaluated by the system [9].

An MRI scan varies from CT scans in that it does not utilize ionizing radiation, which may be dangerous [10].

### 1.3.3 MRI imaging sequences

Anatomy may be seen in three planes with MRI: axial, sagittal, and coronal (see Figure 1.3). By changing an acquisition parameter, such as the time of re-petition between two excitations (TR) or the time between the excitation signal and the echo reception (Echo Time, TE). All of the criteria are chosen based on the clinical investigation that will be conducted. Therefore, different types of sequences are generated. Where, each sequence has its own specification according to the parameter used such as T1, T2, proton density, and so on.

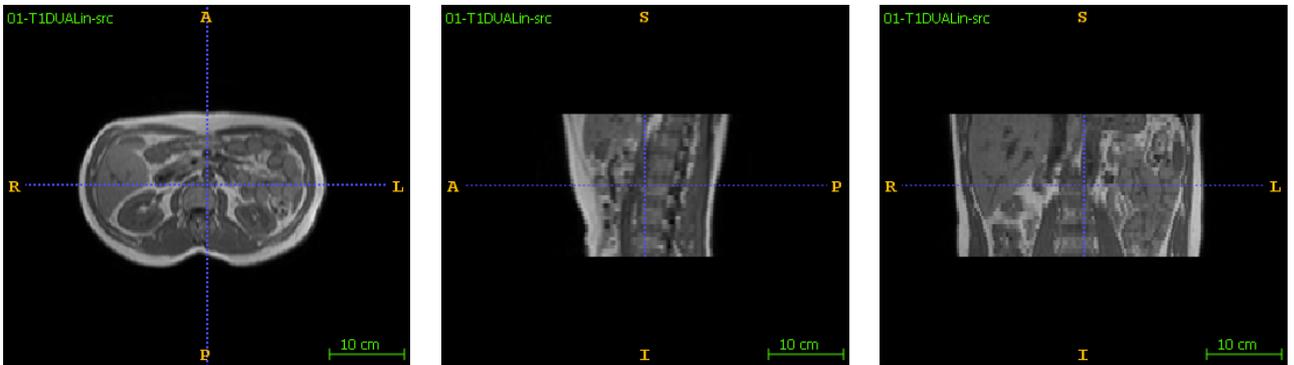


Figure 1.3 – abdominal MR images Planes: Axial, Sagittal, Coronal.

#### 1.3.3.1 T1-weighted Image

Short TE and TR timings are used to create these effects. They are believed to be the best in all MRI protocols because they provide the most anatomical images, which closely resemble the appearances of tissues [11]. For TR of the order of 600 ms, the contrast between the tissues depends essentially on their magnetization speed. For TE of about 20 ms, differences in signal decay between tissues do not have time to express themselves [12]. For example see Figure 1.4 (a).

#### 1.3.3.2 T2-Weighted Image

Longer TE and TR times are used to create these effects. Signal decay dominates the difference in proton density across tissues for long TRs (about 2s) and long TEs (around 90ms), which is weaker, but extremely instructive on tissue composition (mostly moisture content) [12]. For example see Figure 1.4 (b).

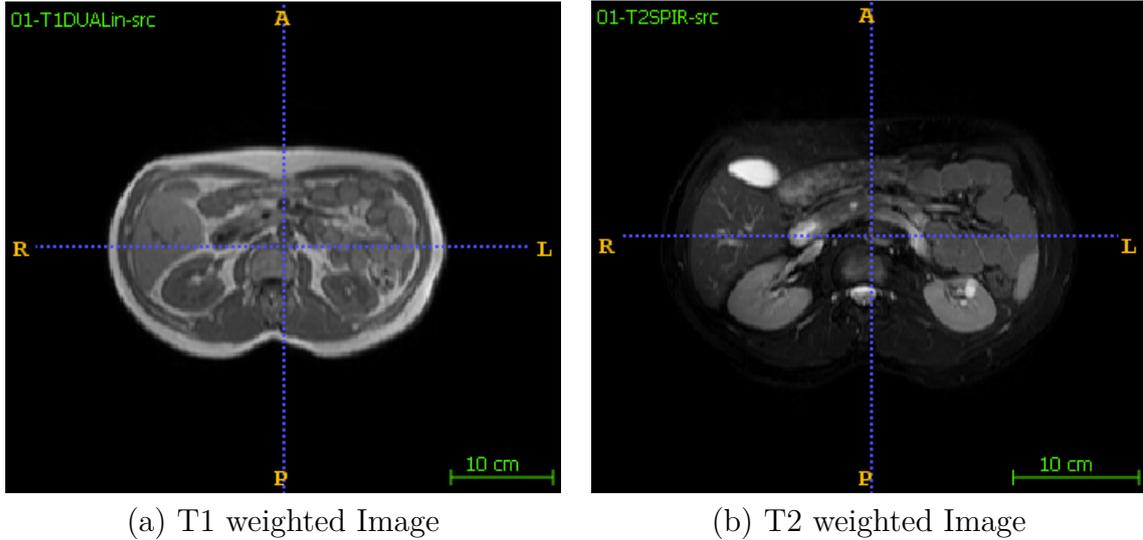


Figure 1.4 – Illustration of examples of abdominal MR images sequences.

## 1.4 Human Abdominal Anatomy

The abdomen (also known as the belly) is the part of the body that is between the thorax (chest) and the pelvis. The top surface of the abdomen is formed by the diaphragm. The abdomen finishes and the pelvis begins at the level of the pelvic bones (see Figure 1.5).

All of the digestive organs, including the stomach, small and large intestines, pancreas, liver, and gallbladder, are housed in the abdomen. Connecting tissues (mesentery) hold these organs together loosely, allowing them to expand and glide against one another. The kidneys and spleen are also found in the abdomen.

In the following, We will present some important organs from abdomen such as the liver, the kidney, and spleen in order to use them in our segmentation system.

### 1.4.1 Liver

The liver is located in the upper right-hand part of the abdominal cavity, under the ribs. The liver is encircled (and protected) for the most part by the lower circumference of the rib cage. Glisson’s capsule is a thin connective tissue layer that surrounds it. The liver is wedge-shaped because it is conformed to the underside of the diaphragm. There are two surfaces to it: diaphragmatic and visceral [13]. It keeps its place by peritoneal reflections, also known as ligamentous attachments. Although it has many functions, the liver is best known for processing blood, separating waste from nutrients. The liver plays an important role in the natural metabolism of the body. Its mission is that storing glycogen, synthesizing plasma protein, participating in nutrients metabolic, and detoxify. Besides that, it also incorporates clotting-factor concentrates, fibrinogen, thrombin.

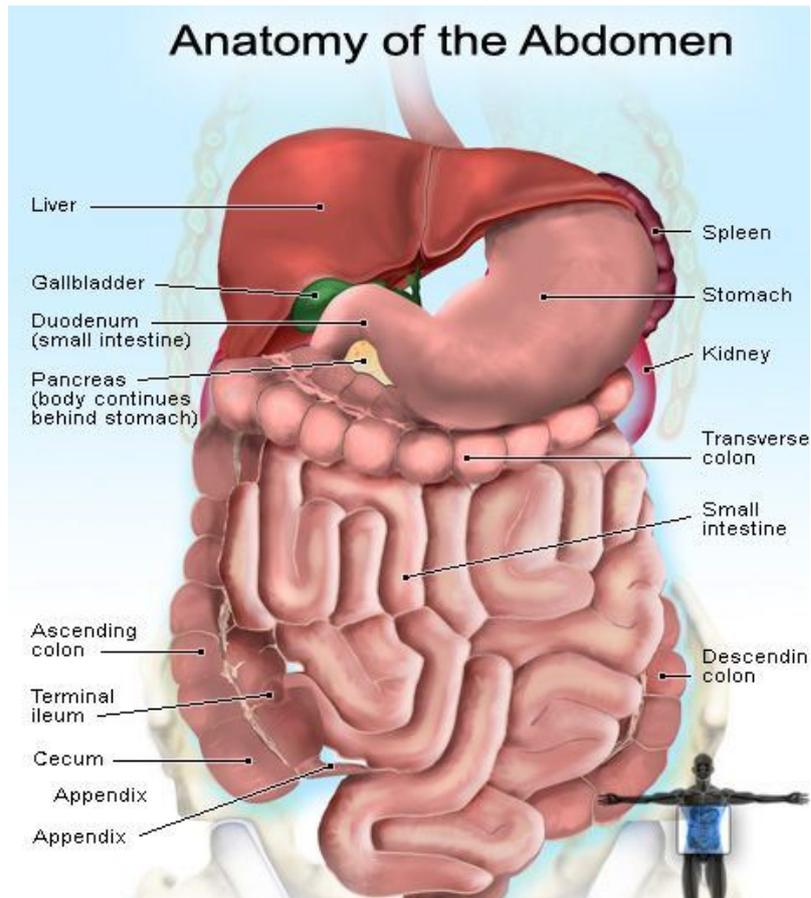


Figure 1.5 – Illustration of human Abdominal organs.

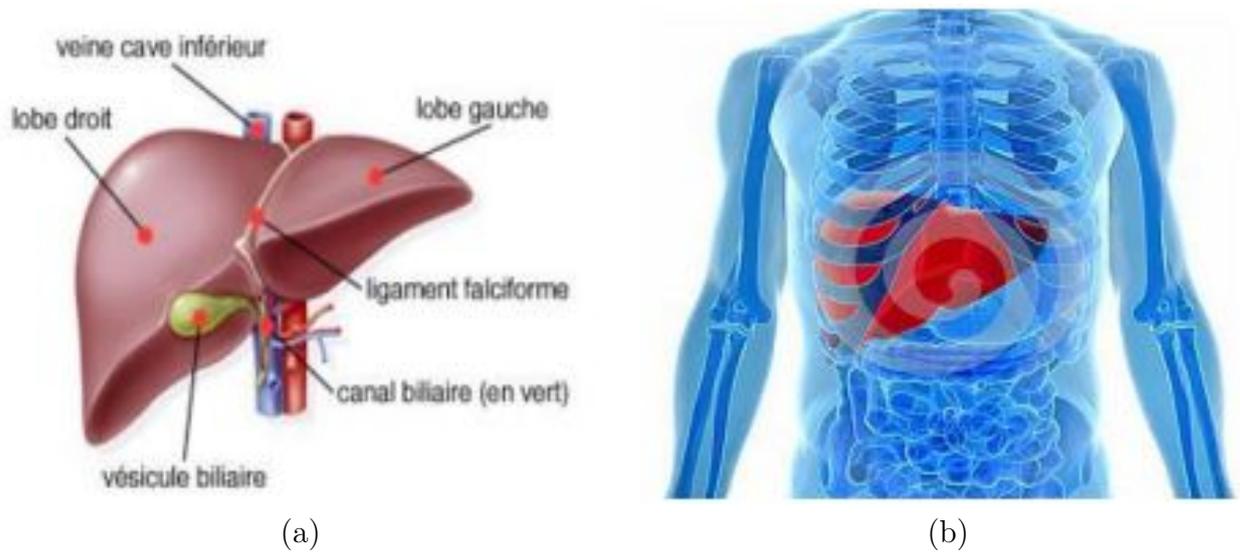


Figure 1.6 – Anatomy of Liver:(a) parts of liver ,(b) location of liver.

### 1.4.2 kidneys

Most people have two kidneys, which are located near the back of the body, under the ribs, on each side of the spine. Because the liver is on the right side of the abdominal cavity, above the right kidney, the left kidney is somewhat higher than the right. which is passed out of the body as urine (see Figure 1.7) . The kidneys also help regulate levels of electrolytes, like salt and potassium, and produce certain hormones that play various roles throughout the body [14].

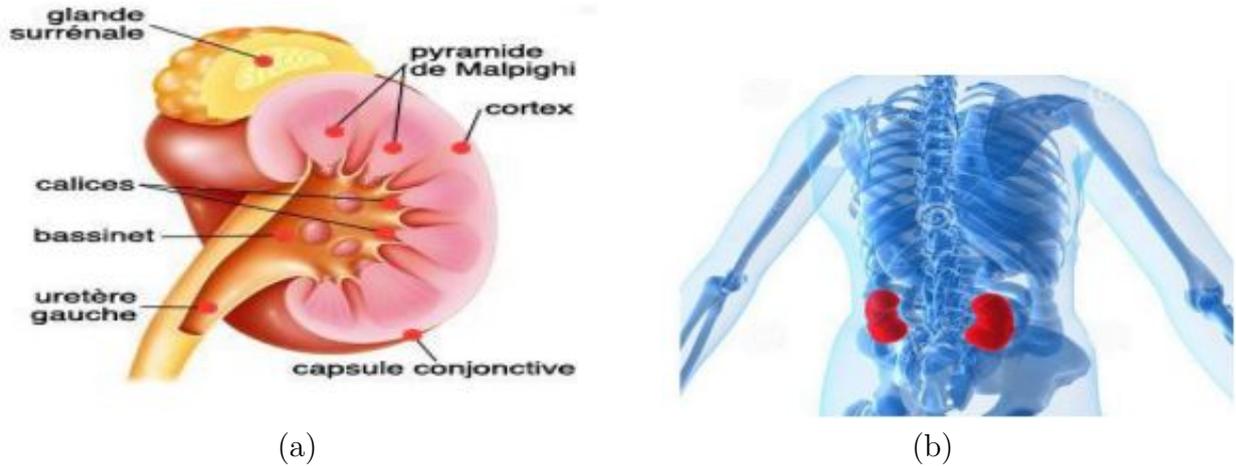


Figure 1.7 – Anatomy of Kidney: (a) parts of kidney ,(b) location of kidney.

### 1.4.3 Spleen

The spleen is an organ located to the left of the stomach in the upper far left region of the abdomen. The size and shape of the spleen varies from person to person, but it's usually fist-shaped, purple, and around 4 inches long. Because the rib cage protects the spleen, you won't be able to feel it until it's extremely big. The spleen has a convex lateral surface that faces the left hemidiaphragm and a reciprocally concave medial surface that is connected to the gastric fundus, pancreatic tail, splenic flexure of the colon, and the anterior side of the left kidney (see Figure 1.8) [15].

## 1.5 Abdominal CT and MRI images segmentation

Generally the medical images segmentation is the most option for the clinics in order to knowledge the parts of body of any patients and who is the problems. Image segmentation is divided into semantic segmentation, instance segmentation and panoramic segmentation. So segmentation of medical images is regarded as a semantic segmentation task.

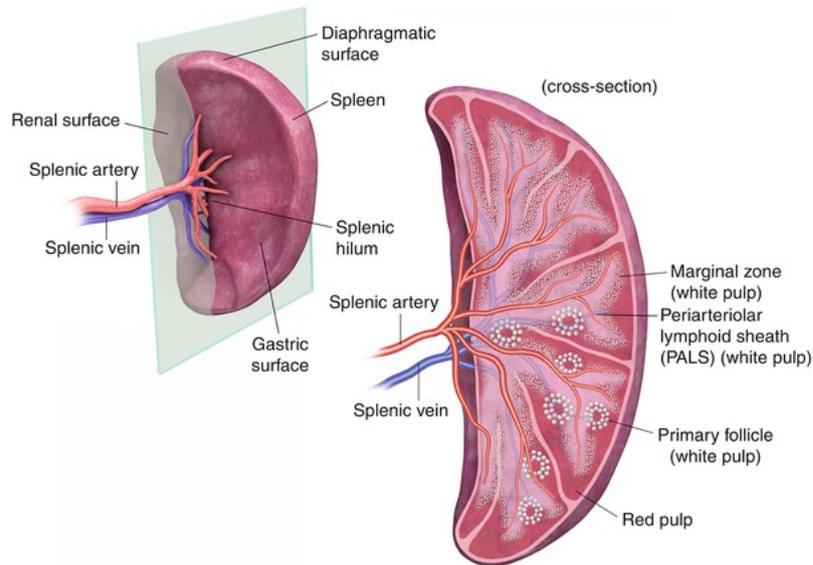


Figure 1.8 – Anatomy of Spleen.

In this section we talk about abdominal CT and MRI images segmentation and the important of this task in diagnosis of abdomen diseases. The process of medical image segmentation can be divided into the following stages [16]:

1. Obtain medical imaging data set, generally including training set, validation set, and test set. When using machine learning for image processing, the data set is often divided into three parts. Among them, the training set is used to train the network model, the verification set is used to adjust the hyper parameters of the model, and the test set is used to verify the final effect of the model.
2. Preprocess and expand the image, generally including standardization of input image, perform random rotation and random scaling on the input image to increase the size of the data set.
3. Use appropriate medical image segmentation method to segment the medical image, and output the segmented images.
4. Estimation performance evaluation. In order to verify the effectiveness of medical image segmentation, effective performance indicators need to be set to be verified. This is an integral part of the process.

### 1.5.1 Abdominal CT and MRI images segmentation approaches

we classified the segmentation techniques in two main parts. Firstly, we present the traditional or semi-automatic methods based segmentation such as: Thresholding, Region, and Edge based techniques. In the second part we have automatic methods of segmentation which are machine learning and deep learning.

### 1.5.1.1 Semi-automatic methods

- **Thresholding:** it is one of the most basic and extensively used picture segmentation techniques which used to convert a greyscale image to a binary image, this approach uses a threshold value.

There are two types of threshold segmentation techniques: global threshold and local (adaptive) threshold [17].

- In the global threshold, a single threshold value is used in the whole image.
- In the local one, a threshold value is assigned to each pixel to determine whether it belongs to the foreground or the background pixel using local information around the pixel.

Obtaining a binary picture first decreases the data's complexity and simplifies the recognition and classification procedure. but, Only two classes are created in the basic version, therefore it cannot be used to multichannel images.

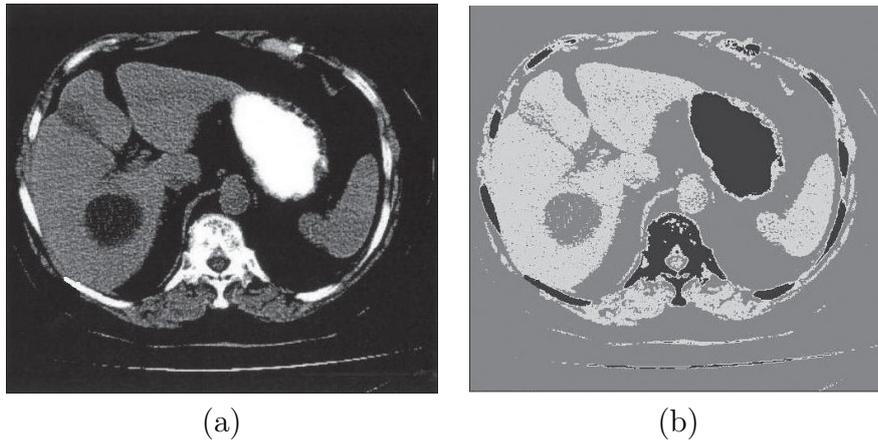


Figure 1.9 – Abdominal segmentation using thresholding method: (a) original CT image, (b) Results of segmentation.

- **Region based methods:** they are based on regions are based on consistency. These algorithms split the entire image into subregions according on certain restrictions, such as the requirement that all pixels in one region have the same gray level. Common patterns in intensity levels within a cluster of surrounding pixels are used in region-based approaches (see Figure 1.10). The area is the cluster, and the segmentation algorithm's purpose is to group the regions according to their anatomical or functional responsibilities [18]. In this technique many types for utilisation such as:
  - **Region Growing:** one of the most prevalent approaches has been region expanding. This approach begins with a pixel and continues to add pixels to the region depending on their resemblance. When a region's expansion ceases, a new seed pixel is picked that does not belong to any other region, and the process is

restarted. The method is continued until all pixels are assigned to a certain location. This approach has the benefit of ensuring linked zones[18].

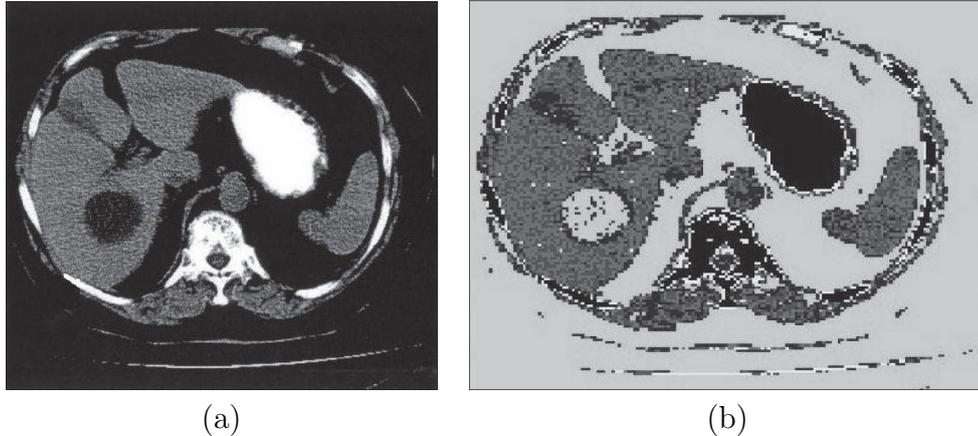


Figure 1.10 – Abdominal segmentation using region based methods:(a) original CT image ,(b) segmentation results.

- **Edge or boundary based methods:** using the variations in grey tones in the images, edge detection algorithms turn images to edge images. Edges indicate a loss of continuity and the beginning of something new. Objects are made up of a variety of pieces in various shades of color. Edges are localized variations in image intensity that appear at the intersection of two areas. The types of Edge based techniques are [19]:
  - Step edge: the visual intensity quickly shifts from one value on one side of the discontinuity to another value on the other.
  - Line Edges: the image intensity changes rapidly, yet after a short distance, it recovers to the original value.
  - Ramp Edges the reality of step edges and Roof edges the reality of line edges

### 1.5.1.2 Automatic Methods

- **Machine learning based methods:** machine learning is a sort of artificial intelligence approach that can find meaningful information from large datasets automatically [20]. This task have two different models, the first one is the traditional model such as the support vector machine (SVM) and k-nearest neighbor (KNN). Which, are also called shallow model. The second one is deep learning model. In fact, there are two main types of machine learning: supervised and unsupervised learning. For shallow model in supervised learning they have for example Artificial Neural Network (ANN) ,Support Vector Machine (SVM) ,and K-Nearest Neighbor (KNN).in addition unsupervised they have K-means Clustering. For deep learning model in supervised learning they have for example deep neural networks(DNN), convolutional

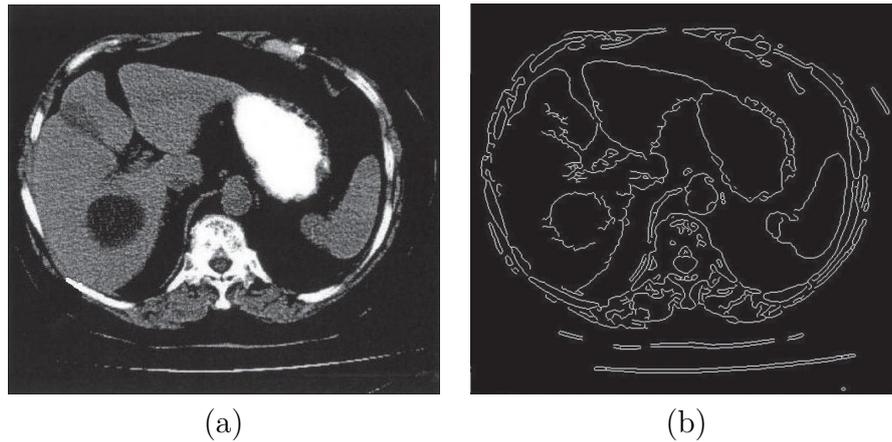


Figure 1.11 – Abdominal CT image:(a) original CT image ,(b) CT image using edge methods .

neural networks (CNN), and recurrent neural networks (RNN).while autoencoders, restricted , and generative adversarial networks (GANs) are unsupervised learning models.

— **shallow model:**

- **Artificial Neural Networks:** an artificial neural network (ANN) is designed to operate similarly to human brains. An ANN is made up of numerous hidden layers, as well as an input layer and an output layer. Units in neighboring strata are fully connected. An ANN has a large number of units and can theoretically approximate any function, hence it can suit nonlinear functions well. Training ANNs is time-consuming because to the complicated model structure. Backpropagation, which cannot be utilized to train deep networks, is used to train ANN models [20]. Artificial neural network (ANN)-based algorithms have been employed for region-based segmentation in a variety of medical imaging, including the segmentation of abdominal CT and MRI organs [11].
- **Support Vector Machine (SVM):** it achieves categorization by creating an N-dimensional hyper plane that divides the input into two groups as efficiently as possible. SVM models and neural networks have a lot in common. In reality, a sigmoid kernel function SVM model is the same as a two-layer perceptron neural network. The goal of SVM modeling is to find the best hyper plane for separating vector clusters so that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other [21]. SVMs are able to solve linear problems well.
- **k-nearest neighbor:** the K-Nearest Neighbors (KNN) classification algo-

rithm assigns samples (image voxels) to a class (organ) based on a search for samples in a learning set with similar attributes. The feature space is filled with the learning set formed from the labeled voxels based on the feature values of its samples. Inserting a new image voxel in the feature space and inspecting the  $K$  learning samples that are closest in a distance measure  $d$  to it classifies it. Then, using a voting procedure among the organs allocated to the  $K$  training voxels, the organ label is assigned to the target voxel [8]. the KNN is apply to massive data ,Suitable to nonlinear data and the Train is quickly in this model but the problem is Low accuracy on the minority class and they have Long test times.

- **K-Means:** the K-Means algorithm based on dividing, is a kind of cluster algorithm. This algorithm which is unsupervised is usually used in data mining and pattern recognition. Aiming at minimizing cluster performance index, square-error and error criterion are foundations of this algorithm. and has advantages of briefness,efficiency and celerity [22]. The unsupervised methods are commonly employed in CT and MRI brain and abdomen image segmentation classification and segmentation. The issue with this method is that it is not noise-resistant. Furthermore, segmentation with FCM is decided only by the intensity of pixels, with no regard for the idea of neighborhood [11].
- **Deep Learning:** in the growth of machine learning and artificial intelligence, deep learning is a research trend. It employs deep neural networks to perform the human brain’s learning process and extract features from massive amounts of input (music, text, photos, etc.) in an unsupervised manner [23].

## 1.6 Conclusion

In this chapter, we gave a brief introduction about the anatomy of the human abdomen . The characteristics of abdominal CT and MR images, such as features and sequences, are also discussed. After that, we offered a brief overview of many classical and conventional approaches (machine learning) for segmenting abdominal CT and MRI organs.

As a result of the complexity of CT and MRI images, the conventional procedures for segmenting organs in abdominal MR images were insufficient. When dealing with vast amounts of data, deep learning algorithms provide astounding outcomes. They have been identified as a source of interest in medical images segmentation studies.

In the following chapter, we will go through how we used deep learning models to segment our multi-organ abdomen CT and MR images.

# Chapter 2

## Deep learning

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>17</b>
<b>2.2</b>	<b>Machine learning (ML)</b>	<b>17</b>
2.2.1	Types of Machine learning Algorithms	17
2.2.2	Artificial Neural Networks	21
<b>2.3</b>	<b>Convolution Neural Networks (CNNs)</b>	<b>22</b>
2.3.1	The Basics of Convolutional Neural Network	23
2.3.2	Activation Functions	24
<b>2.4</b>	<b>U-Net network</b>	<b>25</b>
<b>2.5</b>	<b>Deep Models Matter</b>	<b>27</b>
2.5.1	Visual Geometry Group (VggNet)	27
2.5.2	Residual Network (ResNet)	27
2.5.3	Densely Connected Network (DenseNet)	27
<b>2.6</b>	<b>Advanced Training Techniques</b>	<b>27</b>
2.6.1	Data Augmentation	28
2.6.2	Regularization	28
<b>2.7</b>	<b>Evaluation Metrics</b>	<b>28</b>
<b>2.8</b>	<b>Related Works of abdominal multi-organs segmentation</b>	<b>30</b>
<b>2.9</b>	<b>Conclusion</b>	<b>31</b>

---

## 2.1 Introduction

In the previous chapter, we have presented different segmentation approaches proposed for abdominal CT and MR images segmentation. Despite the range of segmentation methodologies, we have found that deep learning approaches are the most effective for this task.

In this chapter, we will go through the Deep Learning technique in depth. We begin by discussing machine learning and its many sorts of algorithms before moving on to deep learning and the techniques employed. The CNN and the U-net networks are two significant deep learning models that we show next. We go over the training procedure in depth in order to prevent the network from being utilized in our operations. Following that, a brief state-of-the-art is offered about the many works generated in organ segmentation in abdominal CT and MRI exams employing deep learning.

## 2.2 Machine learning (ML)

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms in order to imitate the way that humans learn, gradually improving its accuracy [24]. the benefits to automatically learn from the concepts and knowledge without being explicitly programmed [25]. ML algorithms have been employed in practically every branch of computer science, as well as a wide range of natural, engineering, and social sciences fields. The number of practical applications is significantly higher. Many sectors, would not have thrived without strong machine learning algorithms [24].

### 2.2.1 Types of Machine learning Algorithms

Machine learning algorithms are classified into taxonomies based on the algorithm's expected outcome. The some of common algorithm types [21]:

#### 2.2.1.1 Supervised learning

when the algorithm creates a function that translates the inputs to the desired outputs The classification issue is a common supervised learning challenge in which the learner must learn (or estimate the behavior) of a function that maps a vector into one of many classes by studying multiple input-output samples of the function [21]. Supervised learning is the most common technique for training neural networks and decision trees. The process of applying supervised ML to a realworld problem is described in Figure 2.1 [21].

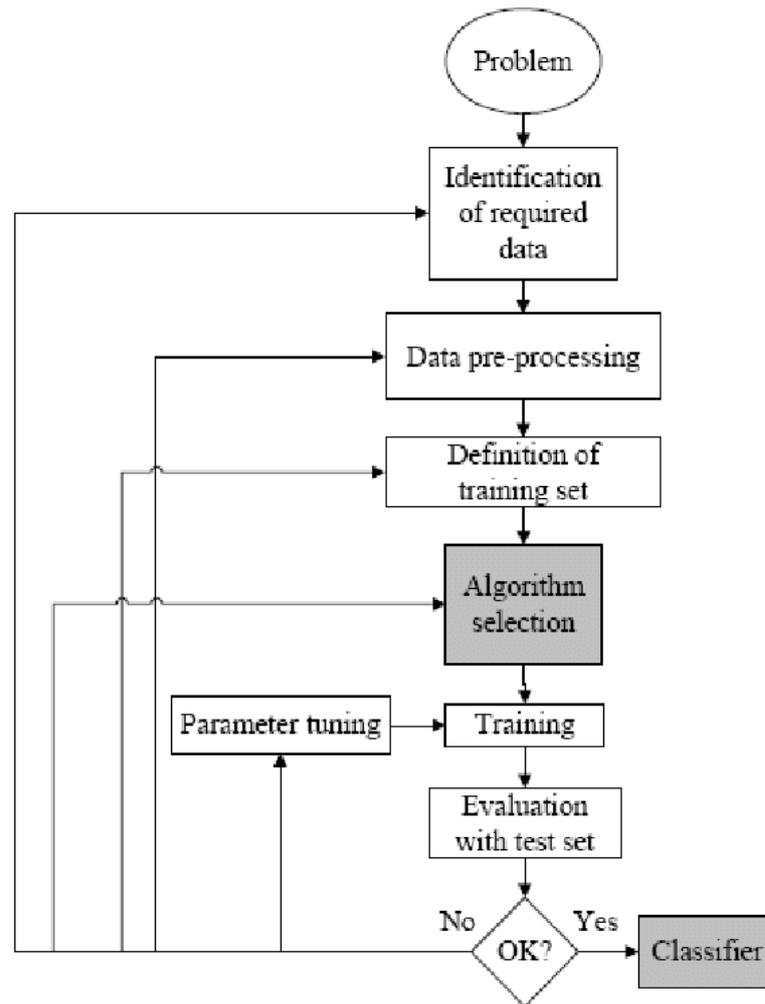


Figure 2.1 – The processes of supervised Machine Learning.

### 2.2.1.2 Unsupervised learning

Unsupervised learning appears to be more difficult, the objective is for the computer to learn something that we don't tell it how to accomplish. Unsupervised learning may be approached in two ways. The first method is to train the agent by utilizing a reward system to signify achievement rather than offering explicit categorizations. Because the aim is to make judgments that maximize rewards rather than provide a classification, this form of training will often fall into the decision problem paradigm. Clustering is a second form of unsupervised learning. The purpose of this sort of learning is to detect similarities in the training data rather than to maximize a utility function. The idea is that the found clusters will fit an intuitive categorisation quite well. When people are clustered based on their demographics, for example, the affluent may be in one group and the destitute in another [21]. Supervised learning often leaves the probability for inputs undefined. This model is not needed as long as the inputs are available, but if some of the input values

are missing, it is not possible to infer anything about the outputs. Unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations is assumed to be at the end of the causal chain. Examples of supervised learning and unsupervised learning are shown in Figure 2.2 [21].

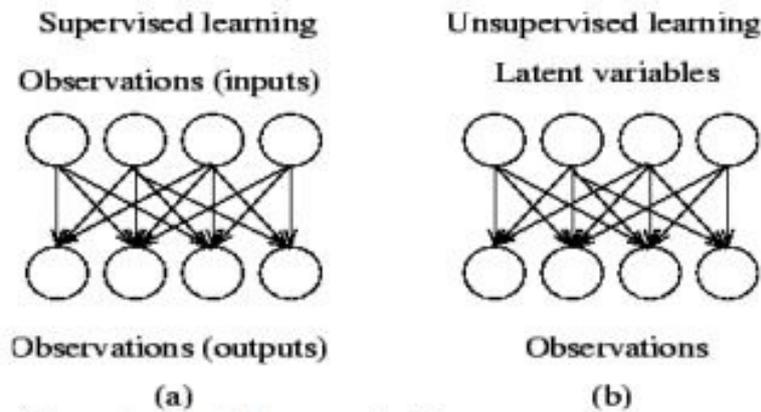


Figure 2.2 – Examples of supervised and unsupervised learning models.

### 2.2.1.3 Semi-supervised Learning

Semi-supervised learning refers to a group of algorithms that attempt to learn from both unlabeled and labeled data drawn from the same or comparable distributions. Different approaches differ in terms of what information may be gleaned from the unlabeled data's structure. The usual evaluation process for semi-supervised learning algorithms is as follows: (1) Begin with a conventional labeled dataset; (2) save just a small percentage of the labels ;and (3) treat the remainder as unlabeled data [26].

### 2.2.1.4 Reinforcement learning

Reinforcement learning is the learning of a mapping from situations to actions as to maximize a scalar reward or reinforcement signal (see Figure 2.3) [27][28]. This merger has provided artificial intelligence researchers with a number of ideas for computer algorithms that develop a strategy that optimizes the agent's long-term return (amount of reward) from completing a job. The two most significant differentiating properties of reinforcement learning are trial-and-error search and delayed reward [28].

### 2.2.1.5 Deep learning

Deep learning is a type of machine learning that allows computers to learn from their mistakes and comprehend the world as a hierarchy of concepts. A graph of these hierarchies would be many layers deep, allowing the machine to understand sophisticated concepts

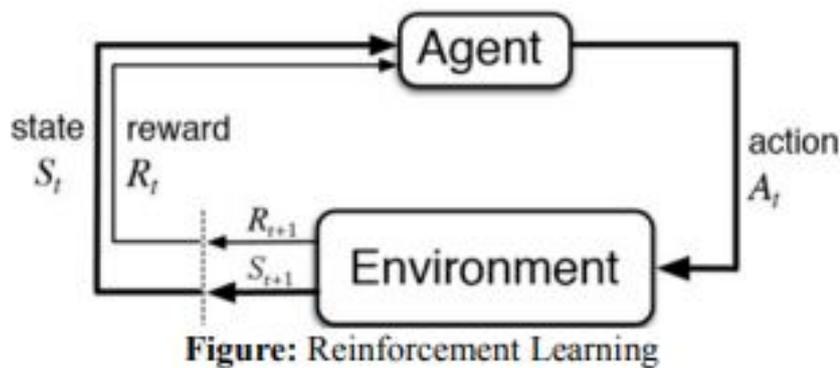


Figure 2.3 – Reinforcement learning.

by constructing them out of simpler ones [29]. There are numerous neurons in a neural network. Each neuron may be thought of as a little data processor. To build the full deep neural network, the neurons are connected in a certain manner. End-to-end image processing is now possible thanks to neural networks. Deep learning is the process of a network’s hidden layers developing into several layers [16].

The artificial neural network system is used in deep learning technologies (ANNs). These ANNs are continually learning algorithms, and the efficiency of training procedures may be increased by continuously increasing the amount of data [25].

The training phase and the inferring phase are the only two steps that make the deep learning process function. The training step entails labeling vast volumes of data and finding their matching properties, whereas the inferring phase entails drawing inferences and categorizing previously unseen data [25]. We summarized the categories of Deep learning methods in the Figure 3.6.

Sequential architectures are common in deep learning models, with the first few layers learning basic properties like edges. For sophisticated object recognition, further layers incorporate these derived characteristics. This characteristic makes DL a good fit for image-related tasks like computer-assisted diagnosis, image enhancement, picture synthesis, and functional information extraction [30].

Deep learning is a wide phrase that encompasses both machine learning and artificial intelligence. Deep learning approaches have reached new heights of success in a range of application areas as a result of the qualities listed below. For example, performance and accuracy have improved in new domains such as decision fusion, on-board mobile devices, transfer learning, class imbalance issues, and human activity detection[31]. Deep learning has the following characteristics:

- The type of the network topology, activation function, and data format all influence

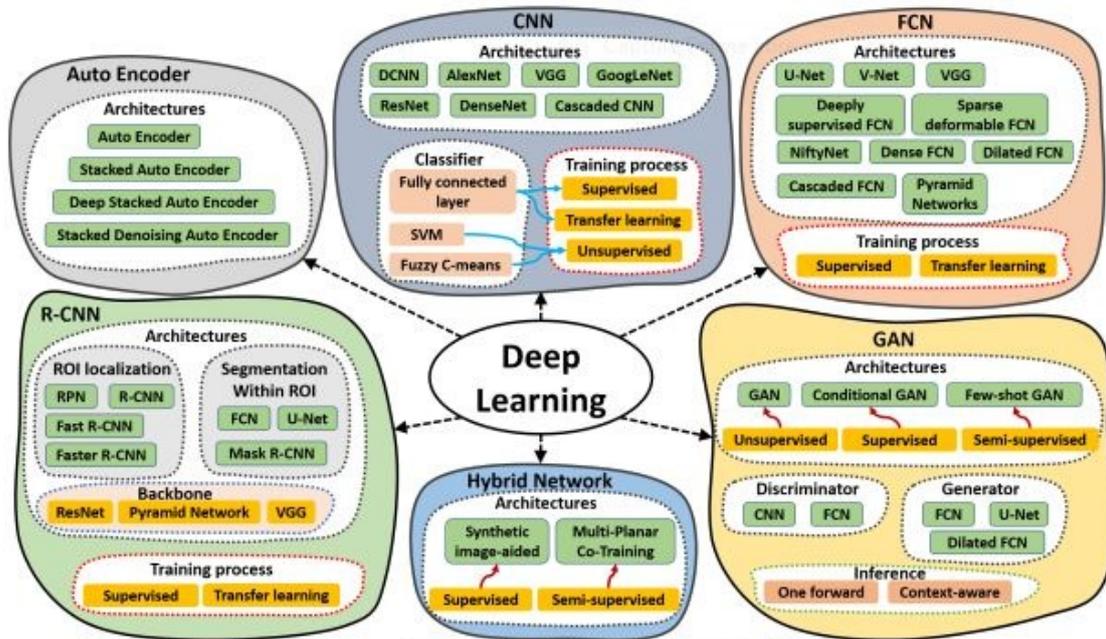


Figure 2.4 – Categories of Deep learning methods.

deep learning networks.

- With little training data and a secure and strong generalization capacity.
- Have a high capacity to learn.
- Is able to make better use of datasets.
- From the data, learn how to extract features.

### 2.2.1.6 Transfer Learning

Transfer learning with pretrained neural network models is a frequent notion in deep learning. When training on a new goal, such as picture segmentation of medical volumes, neural networks that have been trained on another task, such as a natural image classification data set, can be utilized to initialize the network weights. The theory behind this is that the earliest layers of neural networks learn comparable notions to recognize fundamental structures like blobs and edges for various tasks or datasets. When employing pre-trained models, these concepts do not need to be re-trained [32].

## 2.2.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is a mathematical model that tries to simulate the Biological neural networks' structure and functionality are described. Every structure

begins with a basic building block. An artificial neural network is an artificial neuron, that is, a simple mathematical model (function), Multiplication, summation, and activation are three simple laws in this model. The inputs are weighted at the artificial neuron's entry, which implies that each input value is multiplied by its own weight. The sum function in the artificial neuron's center part adds all weighted inputs and bias. The total of previously weighted inputs and bias passes via an activation function, also known as a transfer function, at the exit of an artificial neuron [33].

Artificial neurons receive signals with the values  $x_1, x_2, \dots, x_i$  from other neurons or external sources. The synapse connects and calculates the output  $y$ . Each entry has a weight associated with it. It is assigned based on the relative relevance of the various components. The neuron's  $x$  input value corresponds to the ponderated value of its input. The other component of weight  $b$  is known as the bias. The neuron then applies a  $f$  function to this sum (function of activation) [34].

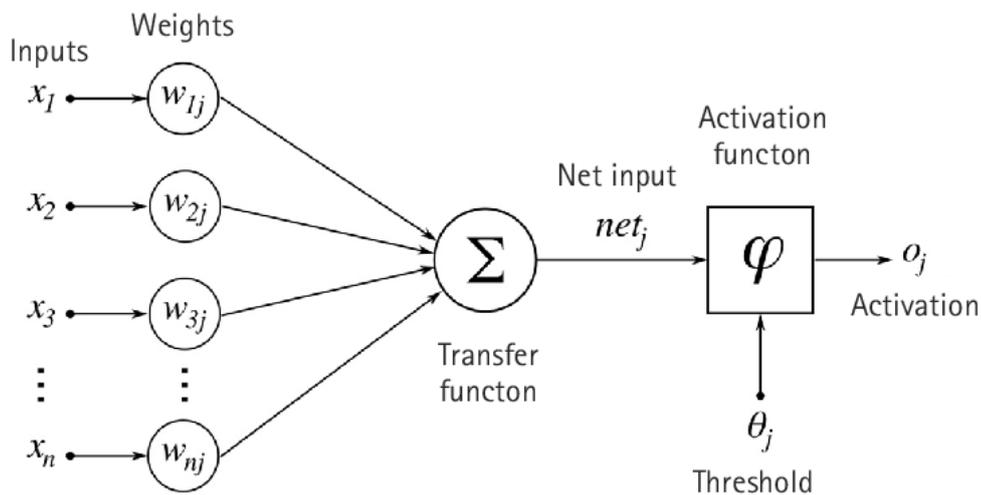


Figure 2.5 – Diagram of the artificial neuron[35].

A neural network with numerous layers of nonlinear processing units is referred to as a deep learning technique. It has seen tremendous evolution in recent years in a variety of sectors, most notably in medical picture segmentation, with a variety of models and architectures, each of which introduces a new approach or improves the network's design. We will analyze two designs, the CNN and U-net networks, among the available architectures and compare their advantages and limitations in order to choose which model will be utilized in our segmentation project.

## 2.3 Convolution Neural Networks (CNNs)

A deep convolutional neural network (CNN) is a form of artificial neural network that needs a convolutional layer but can also include nonlinear, pooling, and fully connected

layers. CNN might be advantageous depending on the application. However, it introduces new training parameters. The backpropagation method is used to train convolutional filters in the CNN. The filter structure's form is determined by the task at hand. For example, in a face detection application, one filter may conduct edge extraction while another performs eye extraction. In CNN, however, we do not have complete control over these filters, and their values are set by learning [34]. CNN is characteristically by does not need feature extraction by manually. So, human expertise are not required to segment tumors or organs in some layouts, and because there are millions of learnable parameters to estimate, is significantly more data hungry and, as a result, more computationally costly, necessitating the use of graphics processing units (GPUs) for model training [36].

### 2.3.1 The Basics of Convolutional Neural Network

A convolutional neural network (CNN) is made up of three different types of computing layers: convolutional, pooling, and fully connected [37].

#### 2.3.1.1 Convolutional Layer

For the given input data, many filters slide across the convolutional layer. The output of this layer is a summation of an element-by-element multiplication of the filters and receptive field of the input. The weighted summation is added to the following layer as an element(show the Figure 2.6) [34].

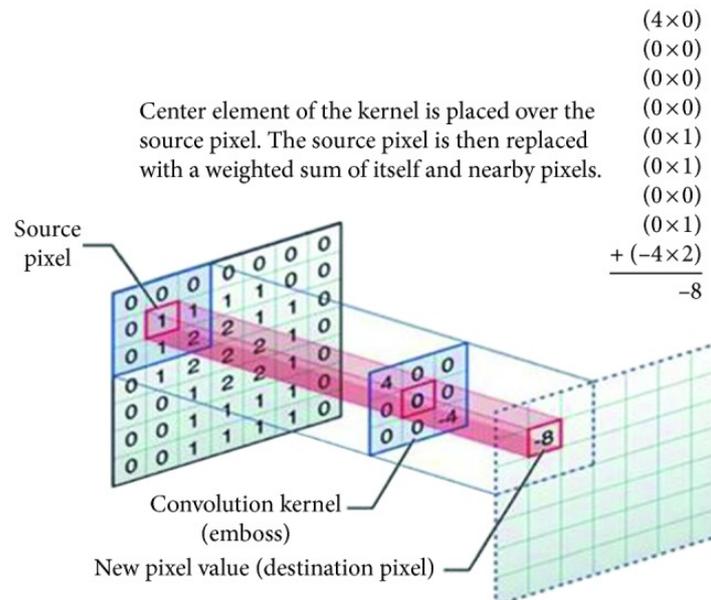


Figure 2.6 – Example of convolutional layer slides [34].

### 2.3.1.2 Pooling Layer

By executing a down-sampling procedure along the spatial dimensions, this layer hopes to reduce the amount of extracted features [38]. The pooling layer works by aggregating a set of data, where the input can be of any kind, including arrays, images, and other sorts of data [39]. We consider three types of pooling layers: min-pooling, average-pooling and max-pooling.

- **Min-pooling**: it takes the minimum element from a block in terms of pooling size
- **Average-pooling**: it consists of taking the average of a pooling block's pooling size (i.e., the number of elements that may be covered by a single pooling operation).
- **Max-pooling**: it represents the maximum element from a block in terms of pooling size is returned by max-pooling layers.

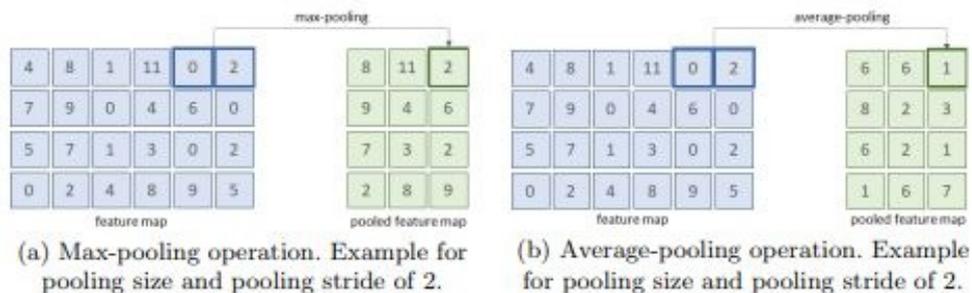


Figure 2.7 – Illustration of different types of pooling operation [38].

### 2.3.1.3 Fully Connected Layer

The final convolution or pooling layer's output feature maps are often flattened, converted to a one-dimensional array of integers (or vector), and linked to one or more fully connected layers. The number of output nodes in the final fully linked layer is usually equal to the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU [36].

## 2.3.2 Activation Functions

### 2.3.2.1 Rectified Linear Unit (ReLU)

The rectified linear unit, or ReLU, is a non-linear activation function commonly employed in neural networks. The advantage of employing ReLU is that not all neurons are

stimulated at the same time. This means that a neuron will only be destroyed when the linear transformation output is zero. Mathematically, it may be defined as:

$$f(x) = \max(0, x) \text{ where } x \text{ is an input value} \quad (2.1)$$

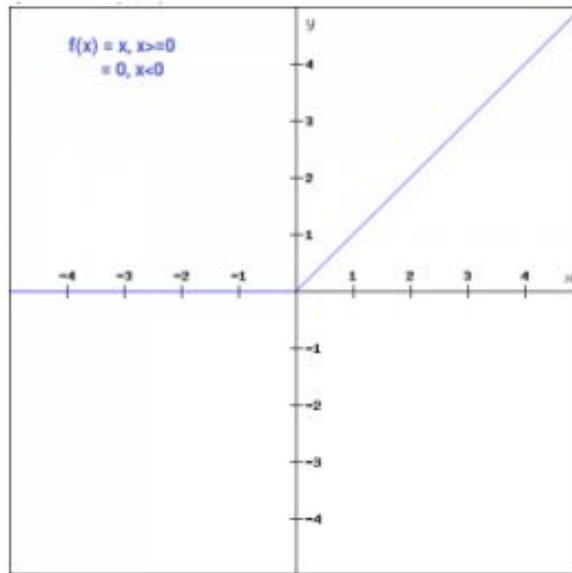


Figure 2.8 – ReLU Activation Function Plot [40].

### 2.3.2.2 SIGMOID

As a non-linear function, it is the most commonly employed activation function. The sigmoid function changes 0 to 1 values. Its definition is as follows:

$$f(x) = 1/e^{-x} \quad (2.2)$$

Sigmoid function is continuously differentiable and a smooth S-shaped function. The derivative of the function is:

$$f(x) = 1 - \text{sigmoid}(x) \quad (2.3)$$

## 2.4 U-Net network

U-Net is an image segmentation neural network that takes an image as input where the outputs are a probability map. U-Net was introduced as a fully-convolutional neural network capable of training on very minimal samples and producing results comparable to

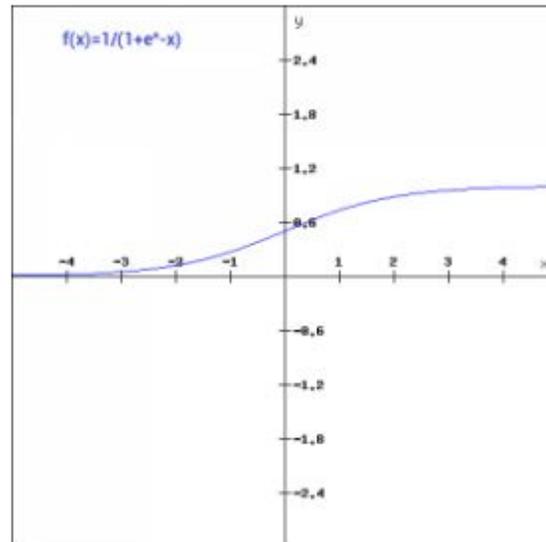


Figure 2.9 – SIGMOID Activation Function Plot [40].

sliding-window models. It outperforms existing approaches on numerous biomedical picture segmentation problems after being trained with data augmentation and improvement techniques [41]. As previously stated, the layers in a CNN network would be completely connected, but in the case of the U-net, they adopt a fully convolutional architecture [8].

The UNet network is divided into two sections: down-sampling and up-sampling. The feature extraction stage of downsampling is where the convolutional and pooling layers are used to extract features from the input picture. The feature map is up-sampled using a deconvolution procedure in up-sampling. This structure of down-sampling and up-sampling is also called a decoder-encoder structure [42].

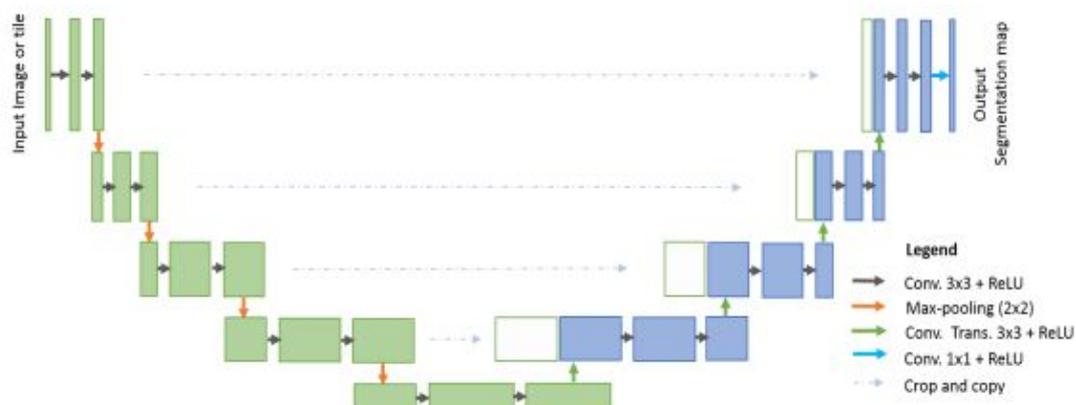


Figure 2.10 – U-Net Architecture [43].

## 2.5 Deep Models Matter

### 2.5.1 Visual Geometry Group (VggNet)

The VGG architecture consists of two convolutional layers both of which use the ReLU activation function. Following the activation function is a single max pooling layer and several fully connected layers also using a ReLU activation function. The final layer of the model is a Softmax layer for classification. In VGG-E the convolution filter size is changed to a  $3 \times 3$  filter with a stride of 2. Three VGG-E models, VGG-11, VGG-16, and VGG-19 [44].

In our study, We are interested to use the VGGNet. We try to define two important models of this VGGNet as follow:

- **VGG16 model:** is composed of convolutions layers, max pooling layers, and fully connected layers. The total is 16 layers with 5 blocks and each block with a max pooling layer.
- **VGG19 model:** is similar to VGG16 but VGG19 has 19 layers with extra convolution layers in the last three blocks. With deep layers.

Both the VGG16 and VGG19 achieve great performance in the image segmentation.

### 2.5.2 Residual Network (ResNet)

ResNet is developed with many different numbers of layers; 34, 50,101, 152, and even 1202. The popular ResNet50 contained 49 convolution layers and 1 fully connected layer at the end of the network [44]. ResNet is a traditional feedforward network with a residual connection.

### 2.5.3 Densely Connected Network (DenseNet)

DenseNet is formed with dense connectivity between the layers rewarding it the name DenseNet. This concept is efficient for feature reuse, which dramatically reduces network parameters. DenseNet consists of several dense blocks and transition blocks, which are placed between two adjacent dense blocks [44].

## 2.6 Advanced Training Techniques

There are different advanced techniques for efficient training of DL approach , such as regularization and Data Augmentation.

### 2.6.1 Data Augmentation

Special care must be taken to avoid overfitting when training large neural networks with minimal training data. We use a wide range of data augmentation approaches to solve this challenge. Random rotations, random scaling, random elastic deformations, gamma correction augmentation, and mirroring were used as augmentation techniques on the fly during training [45]. Data Augmentation have many techniques as follow:

1. **Random erasing:** it is similar to dropout regularization in that it occurs in the input data space rather than in the network architecture, and it is inspired by the mechanics of dropout regularization. This method was created primarily to address picture identification issues caused by occlusion. Random erasing works by randomly selecting an  $n \times m$  patch of an image and masking it with either 0s, 255s, mean pixel values, or random values [46].
2. **Kernel filters:** are a common image processing method for sharpening and blurring pictures. These filters function by sliding a  $n \times n$  matrix across an image with either a Gaussian blur filter or a high contrast vertical or horizontal edge filter, which results in a sharper picture along edges [46].
3. **Rotation:** rotation augmentations are done by rotating the image right or left on an axis between  $1^\circ$  and  $359^\circ$ . The safety of rotation augmentations is heavily determined by the rotation degree parameter. Slight rotations such as between  $1^\circ$  and  $20^\circ$  or  $-1^\circ$  to  $-20^\circ$  could be useful on digit recognition tasks, but as the rotation degree increases, the label of the data is no longer preserved post-transformation [46].

### 2.6.2 Regularization

It refers to a situation where a model learns statistical regularities specific to the training set, i.e., ends up memorizing the irrelevant noise instead of learning the signal, and, therefore, performs less well on a subsequent new dataset. Overfitting can be effectively reduced by regularization.

There are different regularization approaches that have been proposed in the past few years for deep CNN. The simplest but efficient approach called dropout. In Dropout a randomly selected subset of activation is set to zero within a layer [44].

## 2.7 Evaluation Metrics

Many measures are used to evaluate the performance of deep learning models in segmentation tasks. These criteria are used to choose the best models. The following are a few of them:

1. **The accuracy:** is another metric, which is defined as the percentage of true occurrences (both positive and negative) found among all instances found. Precision and inverse precision are weighted arithmetic means of accuracy. It can also be high but with poor precision, implying that the system works effectively but the findings are slightly skewed. Compare this to striking the bullseye, which implies both great accuracy and precision; for additional information [47], see the equation :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

2. **The dice similarity coefficient (DSC):** it was the primary validation statistic for the spatial overlap index. The DSC calculates the amount of overlap between two segmentations,  $A$  and  $B$  target areas [48], and is defined as:

$$DSC(A, B) = \frac{2(A \cap B)}{A + B} \text{ where } \cap \text{ is the intersection} \quad (2.5)$$

3. **volumetric similarity (VS):** it is a similarity metric that takes the volume of the segments into account. Specifically, the absolute volume difference divided by the total volume difference. Volumetric Similarity (VS) is defined as  $1 - VD$ , where  $VD$  is the volumetric distance [49], see the equation :

$$VS = 1 - \frac{|FN - FP|}{2TP + FP + FN} \quad (2.6)$$

Where TP (true positive) refers to a sample that the model properly predicts as positive. When the model properly predicts a sample as being in the negative class, it is referred to as TN (true negative). The term FP (false positive) refers to when a model mistakenly classifies a sample as positive. When the model mistakenly forecasts a sample as being in the negative class, it is referred to as FN (false negative).

4. **loss function:** is the function that computes the distance between the current output of the algorithm and the expected output. It's a method to evaluate how your algorithm models the data. It can be categorized into two groups. One for classification (discrete values, 0,1,2...) and the other for regression (continuous values) [50].

We can defined many loss functions ,so we will talking about some functions as follow:

- **The cross-entropy:** is a class of Loss function most used in machine learning because that leads to better generalization models and faster training.As some types of cross-entropy is the binary cross-entropy that's for binary classification problem, the range value for this class of Loss function (0.002) as a perfect probability , and Log-Loss is the Binary cross-entropy up to a factor  $1 / \log(2)$ .

This loss function is convex and grows linearly for negative values (less sensitive to outliers) [50].

- **Mean Square Error Loss (MSE):** It's the square difference between the current output  $y_{\text{pred}}$  and the expected output  $y_{\text{true}}$  divided by the number of output [50].

## 2.8 Related Works of abdominal multi-organs segmentation

Precise segmentation of abdominal organs is important for a variety of clinical operations, including visual assistance to diagnosis, comprehensive examination of abdominal organs for proper positioning of a transplant before, abdominal aortic surgery, and many more. Ongoing research aims to enhance segmentation findings and overcome several hurdles caused by the abdomen's extremely variable anatomical qualities and modalities' limits as represented in image characteristics.

For instance, Bobo et al(2018) [51] applied the approach to segmentation of the whole abdomen from magnetic resonance imaging sequences (MRI). The authors show that fully convolutional neural networks (FCN) improve abdominal organ segmentation significantly when compared with multi-atlas methods. The FCN they used resulted in a dice similarity coefficient (DSC) of 0.930 for the spleen, 0.730 for the left kidney, 0.780 for the right kidney, 0.913 for the liver. Weston et al(2019) [52] in this work they develop and evaluate a fully automated algorithm for segmenting the abdomen by used à imaging modality CT based on the U-Net architecture , the results of this model calculating by dice scores for the mean (0.98).Chen et al(2020) [53] they proposed a convolutional neural network (CNN) on MR images of 102 subjects for Ten OARs were studied (liver, spleen, pancreas, left/right kidneys, stomach, duodenum, small intestine, spinal cord, and vertebral bodies), and for was measured they used volume overlapping and surface distance , their model given a results with dice scores between (0.87-0.96). Conze et al(2020), [54] also worked in the segmentation with cascaded convolutional and adversarial deep networks. They used two imaging modalities (CT(segment liver), MR (segment liver ,right kidney, left kidney, spleen)) with T1-DUAL and T2-SPIR sequences for the segmentation multi-organs. In fact , they utilized three contributions (deeper models wich is Vgg16-Unet and Vgg19-Unet. Encoder pre-training and cascaded architecture). The obtained results in this work are provided using the dataset CHAOS, reaches the best dice (97%) in CT images. Furtado, Pedro(2021) [55] show in this work has an important impact on research and practical application of DL because it describes how to post-process, quantifies the advantages, and can be applied to any DL approach such as The U-Net, DeepLabv3 and FCN.

These approaches reviewed focused mostly on architectural variations, multiple views,

ensembles and voting in order to improve the quality of segmentation. We conclude that an important improvement can be obtained by applying Unet, Vgg16-Unet and Vgg19-Unet architectures. They are giving best results comparing to other approaches . And also the important point is the excellent performances in the medical image segmentation field.

## **2.9 Conclusion**

In this chapter, we covered the fundamentals of machine learning methodologies, neural network architecture, deep learning fundamentals, and the training phase of these models. Then, we explained two major deep learning architectures in medical picture segmentation, CNN and U-net networks, to help us choose which one to use for our purpose. Furthermore, we have included a brief overview of the state of the art, which includes relevant publications on new and improved methods for segmenting abdominal organs. We conclude that owing to the results achieved by U-net model in medical image segmentation , and because of its efficient with the small database training. we choose the U-net network with two deeper versions vgg16-unet and vgg19-unet, to perform our CT and MRI abdominal organs segmentation.

So, in the next chapter, we will present the conception and the implementation tools of our application and also the obtained results.

# Chapter 3

## Design, Implementation and experimental results

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>33</b>
<b>3.2</b>	<b>Design of Abdominal Multi-organ Segmentation using Deep learning</b>	<b>33</b>
<b>3.3</b>	<b>Global design</b>	<b>33</b>
<b>3.4</b>	<b>Detailed design</b>	<b>34</b>
3.4.1	Dataset preparation	35
3.4.2	Training phase of our deep model	36
<b>3.5</b>	<b>Implementation</b>	<b>38</b>
3.5.1	Environments and developing tools	38
3.5.2	Details informations of used Image dataset	40
3.5.3	Implementation details	41
<b>3.6</b>	<b>Experimental Results</b>	<b>48</b>
3.6.1	The obtained segmentation results of the liver organ	49
3.6.2	The obtained segmentation results of spleen organ in MRI database	53
3.6.3	The obtained segmentation results of right kidney organ in MRI database	57
3.6.4	The obtained segmentation results of Left Kidney in MRI database	60
3.6.5	Results Comparison	64
3.6.6	Discussion	65
<b>3.7</b>	<b>Conclusion</b>	<b>65</b>

---

## **3.1 Introduction**

In this chapter, we discuss the comprehensive design development, implementation and results of our MRI and CT abdominal organs segmentation system. The first section presents the overall and detailed design of our segmentation system. Whereas the second is devoted to the implementation specifics, which begin with the software and hardware needed to build the system. Finally, a series of tests are run on the provided data image, followed by a discussion and validation of the results.

## **3.2 Design of Abdominal Multi-organ Segmentation using Deep learning**

The conceptual investigation as well as the design of our system are presented in this part. First, we show the overall architecture of our system, which highlights the key operations. Then, we will provide the detailed one, in which each procedure will be discussed separately.

## **3.3 Global design**

In order to develop our abdominal multi-organs segmentation, the input data of our system is data sets (CT and MRI Data sets) of 3D images, that is also used for abdominal segmentation.

The data set must be loaded and pre-processed before passed to the deep learning model. Then, we substituted it to the architecture of deep learning for the automatic segmented. After, we got the segmentation results. For the final step we set the prediction of our results(see the Figure 3.1).

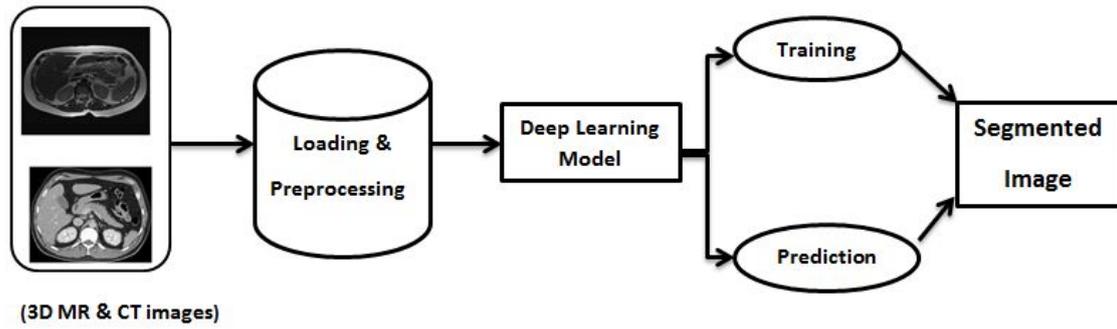


Figure 3.1 – Global design of our abdominal segmentation system.

### 3.4 Detailed design

In this section , we present the different steps of our proposed system. That are starting by the collecting data and pre-processing and then ending with the implementation of our segmentation model (see Figure 3.2).

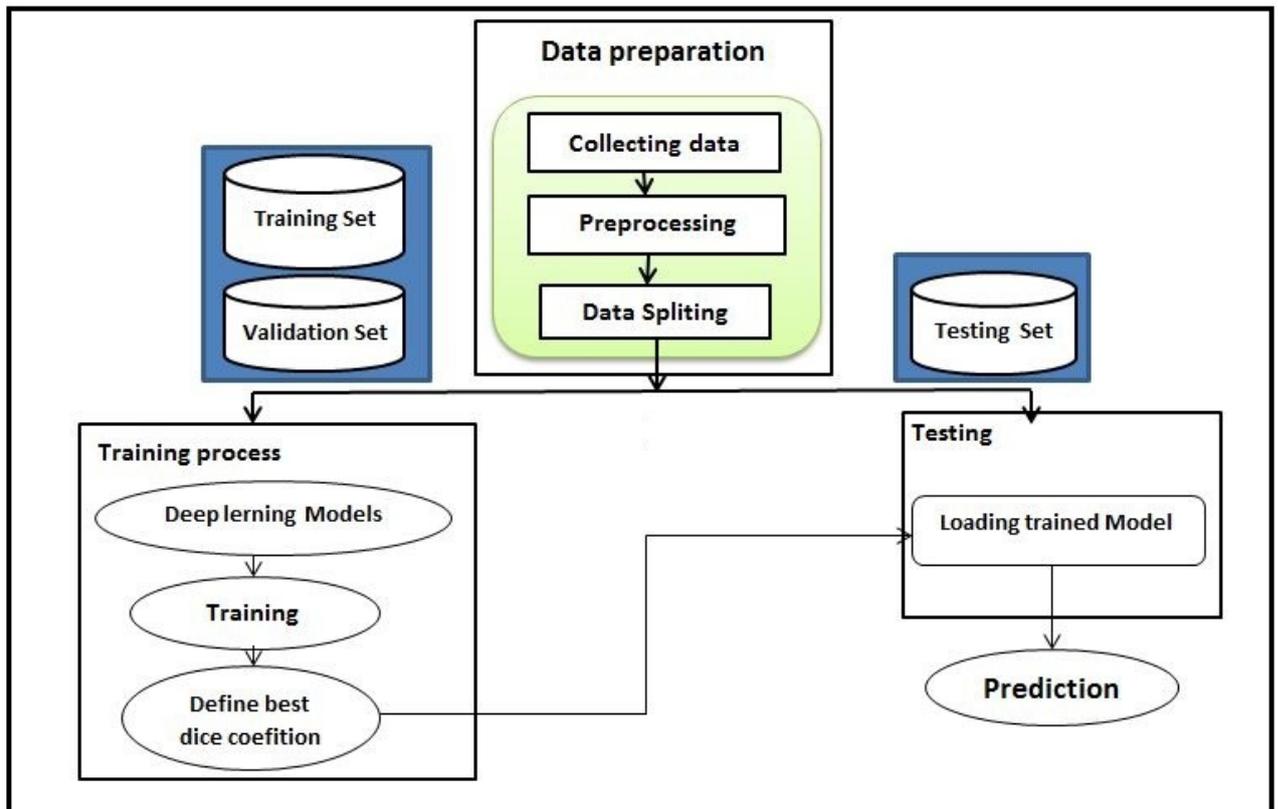


Figure 3.2 – Detailed design of abdominal segmentation system.

### 3.4.1 Dataset preparation

Before feeding the data to the deep learning models, the initial step is to prepare the data. It's broken down into three sub-components.

#### 3.4.1.1 Collecting dataset

In this part we describe the used data base which is offered in the CHAOS challenge [1]. This confirmed original data (scans) was collected from a variety of healthy people. Each patient has a 3D MRI modality with three sequences (T1-DUALin, T1-DUALout and T2-SPIR) in order to segment abdomen into four classes or organs (liver, left kidney, right kidney, spleen). In addition, each patient has a Computed tomographic (CT) scan for use in liver segmentation.

#### 3.4.1.2 Loading and Pre-processing

For the step of data loading, we read the 3D images for all modalities (MRI and CT) then we extracted it to 2D slices for 3D MRI images, and then we applied the pre-processing data which consists of resizing and normalizing the 2D slices and CT images. In the following, we try to explain in detail these treatments:

- **Resizing of data:** it is a process which is applied on all 2D slices of MR images modality in order to give the same dimension ( $256 \times 256$ ). For the CT images modality, we resized it from  $512 \times 512$  to  $64 \times 64$ .
- **Normalization of data:** is used in order to standardize the range of independent variables or features of data, in this method we change the range of pixel intensity values by the famous and easy operation (min-max normalization). It comprises of re-scaling the range of image intensity values to scale the range in  $[0, 1]$  or  $[-1, 1]$  according to the following equation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

#### 3.4.1.3 Data splitting

The most frequent splitting strategy of the dataset, in deep learning, is to divide the dataset into training, validation, and test sets. In our work, for MR Images provided in the CHAOS dataset challenge, we planned to share 50% of the whole dataset for the training set and 50% for the testing set. For CT images we divide the dataset 80% for the training set and 20% for the testing set.

— **Generated training set:**

- For MR Images: We take 50% exams (3D MR images) to generate the training set. In each organ and in the adapted sequence, from each 3D MRI exam, we extract the 2D slices and its corresponded mask that containing the organ selected. Then, we save all the 2D slices and its corresponded masks into Numpy files which represent the input of our deep learning networks.
- For CT images: We select 80% exams (3D CT images) to generate the training set. After resizing and normalizing all 3D CT images and its 2D masks, we save all into Numpy files which represent the input of our deep learning network.

— **Generated testing set:**

- MR images: we use the 50% exams of data. Then, we apply the same processes of the training set to the testing set, starting by extracting the 2D slices and its masks that containing the organ selected, with the modality combination for providing rich testing set, and finishing with saving the Numpy files in order to fit into the model.
- CT images : we utilise the 20% (CT images). then we apply the same methods of the training set for CT images. Then, we save it to Numpy files and done to fit into the models.

### **3.4.2 Training phase of our deep model**

In this step and after the generalization of the training data, we move to the train our deep learning architecture, and fit in the training and validation data (slices and its correspondent masks for both train set and validation set). In this step, multiple iteration or epochs at each time are applied, where, the best generated weights are saved, based on them the model is automatically trained so it will be able to perform the segmentation ( see Figure 3.3).

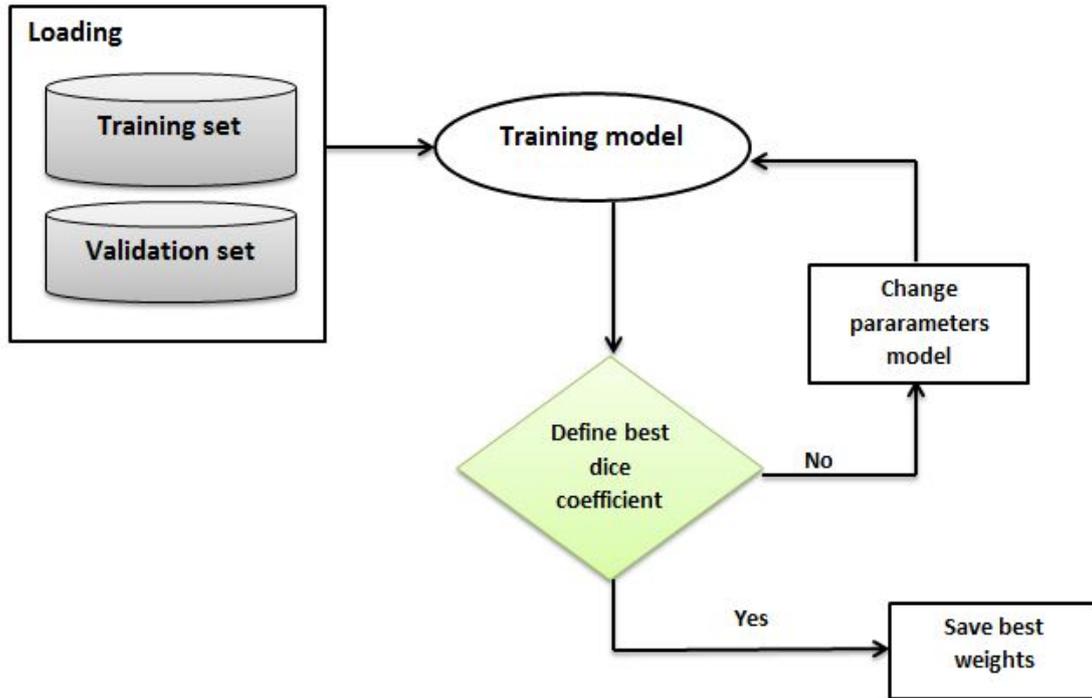


Figure 3.3 – Illustration of training phase of our deep model.

### 3.4.2.1 Prediction phase

We now move on to the prediction step to put our models to the test. We load the model and stored weights from the training stage and run a prediction operation to evaluate if our models is well-trained and performing the job appropriately or not, using evaluation metrics and segmentation visualization results (see Figure 3.4).

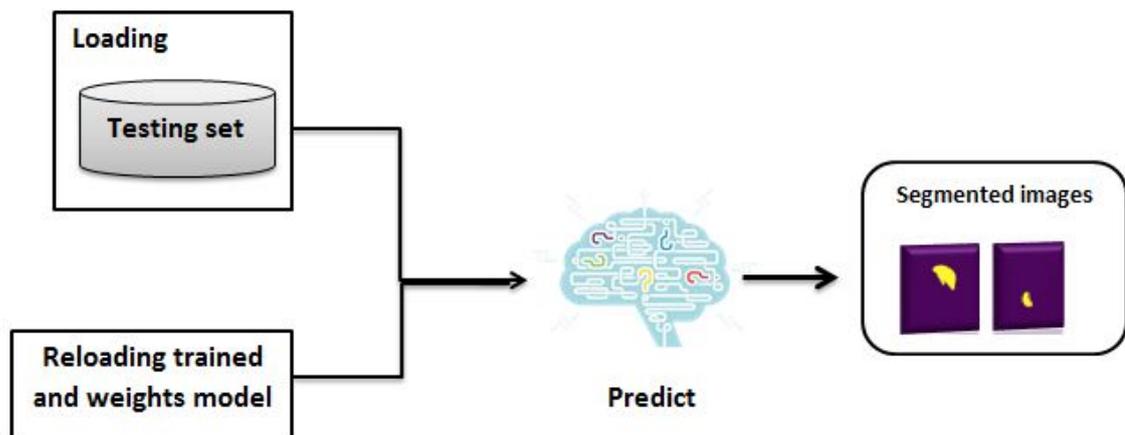


Figure 3.4 – Illustration of Prediction phase.

## **3.5 Implementation**

### **3.5.1 Environments and developing tools**

To implement our abdominal segmentation, we needed various environments, packages, APIs, library's and programming languages.

#### **3.5.1.1 Hardware configuration**

Our hardware configuration is an HP Notebook PC of the following characteristics:

- Processor : Intel(R) CORE(TM)i3 6006U CPU .
- Processor Frequency : 2,00 GHz.
- RAM: 4 Go.

#### **3.5.1.2 Development environment**

For the development of our system, different environments and tools are used, to build our system as showing in the Table [3.1](#):

Tools logo	Descriptions
	<p>Python is a programming language created by Guido van Rossum and first released in 1991. Python is successfully used in thousands of real-world business applications around the world e.g., Google and YouTube. The primary rationale for adopting Python for ML and DL is because it is a general-purpose programming language for research, development and production, at small and large scales. Python features a dynamic type system and automatic memory management, with a large and comprehensive libraries for scientific computation and data analysis [56].</p>
	<p>ANACONDA is an open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system conda [57].</p>
	<p>Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package [58].</p>
	<p>Google Colaboratory (also known as Colab) is a free jupyter notebook environment that runs in the cloud and stores its notebooks on Google drive. Colab was originally an internal Google project; an attempt was made to open source all the code and work more directly upstream, leading to the development of the "Open in Colab" Google chrome extension, but this eventually ended, and Colab development continued internally. As of October 2019, the Colaboratory UI only allows for the creation of notebooks with Python 2 and Python 3 kernels.</p>

Table 3.1 – Illustration of tools environment.

### 3.5.1.3 Packages and APIs

In our project , we used different programming packages and APIs which are helpful for defined any function utilized for building deep learning model (see Table 3.2).

Packages and APIs logo	Descriptions
	<p>is a free and open-source software library for machine learning and artificial intelligence. TensorFlow was developed by the Google Brain team for internal Google use in research and production, it can be used in a wide variety of programming languages, most notably Python, as well as Java script, C++, and Java [59].</p>
	<p>Keras is Python wrapper library that provides bindings to other DL tools such as TensorFlow, CNTK, Theano, beta version with MXNet and announced DeepLearning4j. Keras runs on Python 2.7 to 3.6 and can seamlessly execute on GPUs and CPUs given the underlying frameworks [56].</p>
	<p>NumPy is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data [60].</p>
	<p>OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code [61].</p>
	<p>matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+ [62].</p>
	<p>Scikit-image, or skimage, is an open source Python package designed for image preprocessing, skimage is fairly easy to learn and use.</p>

Table 3.2 – Illustration of packages and APIs.

### 3.5.2 Details informations of used Image dataset

In our study, we use two abdominal databases that contain both CT and MRI (T1 and T2 weighted) provided in the challenge [1]. Where, each data set corresponds to a set of

DICOM images from a particular patient. The data sets are acquired retrospectively and randomly from the PACS of DEU Hospital. The data sets collected from CT and MR databases have no relationship.

The first database contains CT images of 40 different patients. These patients are potential liver donors, who have healthy (no tumors, lesions or any other diseases) liver. Each data set consists of 16 bit images with a resolution of 512x512 [1] (see Figure 3.5).

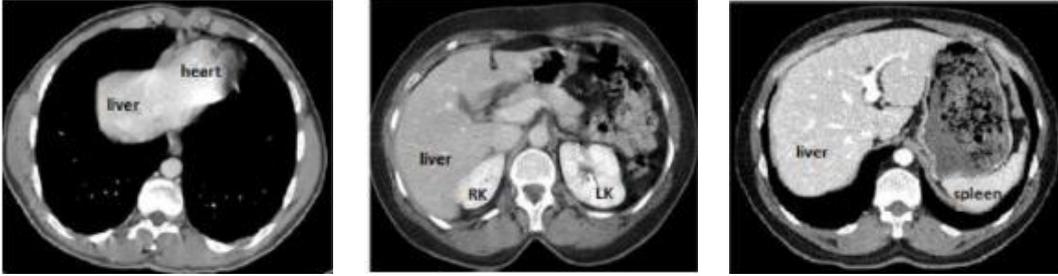


Figure 3.5 – Examples of images from CT database.

The second database includes 120 data sets from two different 3D MRI sequences (T1-DUAL in phase 40 data sets), out phase (40 data sets) and T2-SPIR (40 data sets). This database also does not include any tumors or lesions at the borders of the annotated organs of interest (liver, kidneys, spleen) which produces 12 bit images having a resolution of  $256 \times 256$  (see an example in fig:zc4).

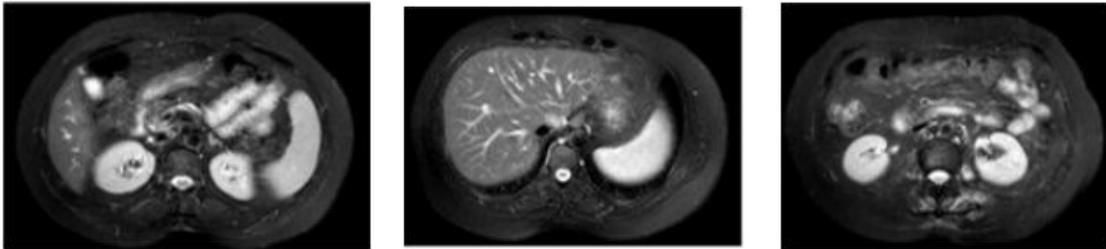


Figure 3.6 – Examples of MR images from database.

### 3.5.3 Implementation details

#### 3.5.3.1 Data preparation and loading

This step allows us for reading and pre-processing our used databases (MRI or CT). Firstly, we start by setting the necessities packages and the modules (as shown in the Figure 3.7). Then, we provide the used codes sources for both MRI and CT databases.

```

import sys
sys.path.append("/content/drive/MyDrive/")
import os.path
import distutils.dir_util
import nibabel as nib
import numpy as np
from skimage.exposure import rescale_intensity
from skimage import io
from skimage.transform import resize
from skimage.segmentation import mark_boundaries
import keras
from keras.models import Model
import tensorflow as tf
from keras.layers import Input, concatenate, Conv2D, MaxPooling2D, Conv2DTranspose
from tensorflow.keras.optimizers import Adam
from keras import backend as K
from keras.applications import vgg16, vgg19
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt

```

Figure 3.7 – Illustration of used packages and modules.

## 1. MRI Database:

- **Reading Data:** In this process, we use a function that allows us to read all 3D MRI data with their corresponds masks if it's existed (train and test data) as showing in Figure 3.8.

```

if modality == 'T1DUALin':
    img = nibabel.load(path+'T1DUALin-src.nii.gz')
elif modality == 'T1DUALout':
    img = nibabel.load(path+'T1DUALout-src.nii.gz')
elif modality == 'T2SPIR':
    img = nibabel.load(path+'T2SPIR-src.nii.gz')

if modality == 'T1DUALin' or modality == 'T1DUALout':
    if os.path.isfile(path+'T1DUAL-mask.nii.gz'): # if mask exists
        mask = nibabel.load(path+'T1DUAL-mask.nii.gz')
elif modality == 'T2SPIR':
    if os.path.isfile(path+'T2SPIR-mask.nii.gz'): # if mask exists
        mask = nibabel.load(path+'T2SPIR-mask.nii.gz')

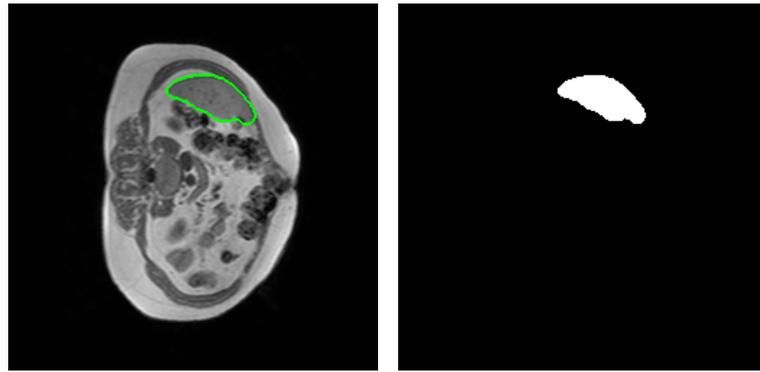
```

Figure 3.8 – Illustration of used function of Reading the MRI data.

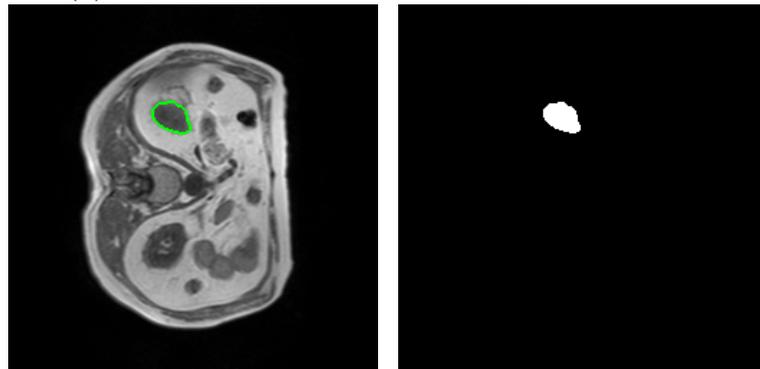
- **Pre-processing data:** We extract the 3D data to 2D slices and resizing these slices with their orrespondent masks into  $256 \times 256$ . Then, we provide in each slice the contours in order to visualize the shape of organs (for example see Figure 3.9). The code function is demonstrated in the Figure 3.10.

```
for id_ in train_ids+test_ids:
    img_, mask_ = train_read_examination(data_folder, id_, modality)
    Zs = np.unique(np.where(mask_.get_data()==organ_label)[2]) # ids of axial slices
    for z in Zs:
        if id_ in train_ids:
            train_count += 1
        elif id_ in test_ids:
            test_count += 1
        img = np.array(img_.get_data()[:, :, z], dtype=np.float)
        _srcs.append(resize(img, output_shape=(256,256), preserve_range=True))
        _mask = mask_.get_data()[:, :, z]
        _mask[np.where(_mask!=organ_label)] = 0
        _mask[np.where(_mask==organ_label)] = 1
        _mask = resize(_mask, output_shape=(256,256), preserve_range=True)
        _mask[np.where(_mask>0.95)] = 255
        _mask = _mask.astype(np.uint8) # "intger"
        _masks.append(_mask)
        _ids.append('exam-%0*d'%(2,id_)+'-slice-%0*d'%(2,z))
```

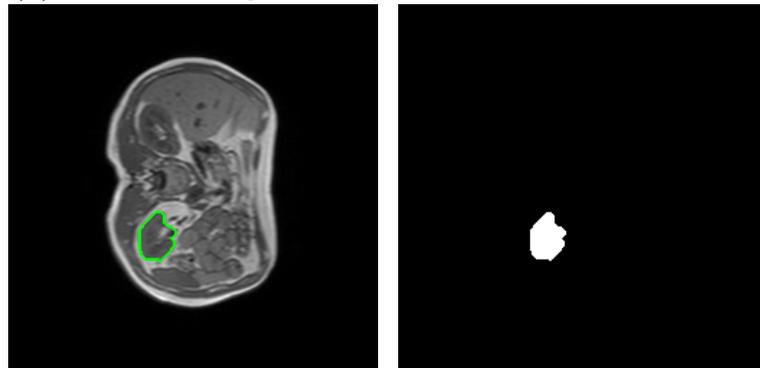
Figure 3.9 – Pre-processing data function.



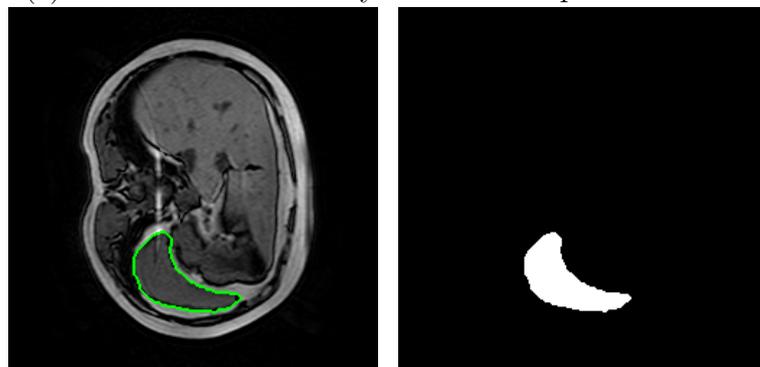
(a) Contour of liver and it correspondent mask



(b) Contour of right kidney and it correspondent mask



(c) Contour of left kidney and it correspondent mask



(d) Contour of spleen and it correspondent mask

Figure 3.10 – Illustration of examples of 2D slices with organs contours and their correspondents masks.

- **Saving the data to Numpy files:** we save all the 2D slices with their masks to Numpy files (see Figure 3.11).

```
np.save(output_folder+'imgs-train.npy', train_imgs)
np.save(output_folder+'imgs-mask-train.npy', train_imgs_mask)
```

Figure 3.11 – Function used to save data on Numpy files.

- **Normalization:** this function include the parts of data preprocessing after the generation of 2D slices with their correspondent masks as showing in the following Figure 3.12.

```
def normalization(imgs, imgs_masks):
    imgs = np.array(imgs).astype('float32', copy=False)
    mean = np.mean(imgs) # mean for data centering
    std = np.std(imgs) # std for data normalization
    imgs -= mean
    imgs /= std
    if type(imgs_masks) == np.ndarray:
        imgs_masks = np.array(imgs_masks).astype('float32', copy=False)
        imgs_masks /= 255.
        imgs_masks = imgs_masks.astype(np.uint8)
        return imgs, imgs_masks
    else:
        return imgs
```

Figure 3.12 – Normalization function.

## 2. CT Database:

- **Loading and pre-processing:** the reading of CT images is differ to the 3D MRI. Therefore, in this part we provide the used code to read the 3D images with their masks. In addition, the normalization of each data is demonstrated (see Figure 3.13).

```
img = pydicom.dcmread(image_path + image)
img=img.pixel_array
img=np.array(img).astype("int16").astype('float32', copy=False)
img = resize(img, (IMG_W, IMG_H))
img = (img - np.min(img)) / (np.max(img) - np.min(img))

dataX[i] = np.expand_dims(img, axis = 2)

for i, image in enumerate(mask_list):
    img = cv2.imread(mask_path + image, cv2.IMREAD_UNCHANGED).astype('float32')
    img = cv2.resize(img, dsize=(IMG_W, IMG_H))
    dataY[i] = np.expand_dims(img, axis = 2)
dataY /= 255.
```

Figure 3.13 – CT Loading and pre-processing functions.

- **Splitting the data:** this step allows to divide our database into training testing sets as showing in the Figure 3.14.

```
trainX, testX = train_test_split(dataX, test_size = 0.2, random_state=50)
trainY, testY = train_test_split(dataY, test_size = 0.2, random_state=50)
```

Figure 3.14 – Function of Splitting the data .

### 3.5.3.2 Creating of our Deep Learning Models

After loading data and saving it to Numpy files, we build our Unet Model with two deeper versions witch are the Vgg16-unet and Vgg19-Unet architectures with it parameters. We defined their layers such as the convolution layers and max-pooling layers. Generally Unet model have two main blocks witch as encoder block and decoder block.

- **Encoder Block:** the encoder architecture will use consecutive inputs starting from the first layer all the way to the bottom. The encoder function as we have defined will have the convolutional block followed by ReLU layers. Once we pass them through the convolution blocks, we will quickly down sample these elements, by using a max-pooling layer and stick to the parameters mentioned in our model as the strides = 2. We will then return both the initial output and the max-pooled output, as we need the former for performing the skip connections [63].
- **Decoder Block:** this block will include three arguments, namely the receiving inputs, the input of the skip connection, and the number of filters in the particular building block. We will upsample the entered input with the help of the Conv2DTranspose layers in our model. Then, we will concatenate both the receiving input and the newly upsampled layers to receive the final value of the skip connections. We will then use this combined function and perform our convolutional block operation to proceed to the next layer and return this output value [63].

### 3.5.3.3 Compiling Model

Once the model is created, we can configure the model using the function *Model.compile()* which takes three arguments: optimizer , loss function and the metrics, we have used:

- Adam [64] optimization algorithm was used to train the network. It consists of a stochastic gradient descent method, combined with a learning rate of 0.001.
- Loss dice coefficient loss function is our loss function which measures the performance of the model segmentation, the output is between 0 and 1.
- Dice coefficient is the adopted metric for evaluate the performance of the model.

### 3.5.3.4 Data augmentation function

In our model we needed to use the function of data augmentation in order to eliminate the over-fitting problem on our results as shown Figure 3.15).

```
batch_size = 6
data_gen_args = dict(rotation_range=20, width_shift_range=0.2, height_shift_range=0.2,
                    shear_range=0.2, zoom_range=0.2, fill_mode='constant',
                    featurewise_center=False, samplewise_center=False, featurewise_std_normalization=False, samplewise_std_normalization=False)
image_datagen = ImageDataGenerator(**data_gen_args)
mask_datagen = ImageDataGenerator(**data_gen_args)
seed = 1
image_datagen.fit(imgs_train, augment=True, seed=seed)
mask_datagen.fit(imgs_mask_train, augment=True, seed=seed)
image_generator = image_datagen.flow(imgs_train, None, shuffle=True, batch_size=batch_size, seed=seed)
mask_generator = mask_datagen.flow(imgs_mask_train, None, shuffle=True, batch_size=batch_size, seed=seed)
```

Figure 3.15 – Illustration of the data augmentation function.

### 3.5.3.5 Training Models

To train models we used the function of Keras package *model.fit\_generator()*. This function enables us to physically fit our data to the model. It takes as input the following parameters (see Figure 3.16):

- **The training data:** both the prepared and processed training set and the masks set.
- **The validation data** or testing data.
- **Number of epochs:** which represents the number of rounds that our model is passing through the data-set.
- **Steps per epoch:** or iteration which represents how many step needed to complete one epoch.
- **Model check point:** callback that is used to save the model, the weights as it shows the metrics during the training.

```
period= 1
model_checkpoint= keras.callbacks.ModelCheckpoint(output_root+'weightsSpleen.h5',monitor='val_dice_coef',
        verbose=1,
        save_best_only=True,
        save_weights_only=True,
        mode='max',
        period=period)
hist = model.fit_generator(itertools.zip_longest(image_generator, mask_generator),
        epochs=250 ,steps_per_epoch=10, validation_data=(imgs_test,imgs_mask_test), callbacks=[model_checkpoint])
```

Figure 3.16 – Illustration of the fitting model function.

### 3.5.3.6 Prediction phase

This phase is provided to apply the predictions on new test images, in order to assess and evaluate the model's performance and learning. The used functions toward the prediction phase are presented as follow:

- **Load the model:** we load the best trained model with the saved weights from the previous stage .
- **Predict:** calling `model.predict()` where it takes as input the test set or the data that we wanted to predict on them , the epochs and the exact label that you want to predict on as well.

### 3.5.3.7 Visualising Results

The first is the plotting of curves that show the evolution of metrics during the learning process (loss and dice coefficients) for both the training and testing stages (see Figure 3.17). The second one is the result of the predicted data (segmented organs) which we generate on 2D image.

```
historique = hist
plt.plot(historique.history['dice_coef'])
plt.plot(historique.history['val_dice_coef'])
plt.title('model loss ')
plt.ylabel('dice')
plt.ylim([0,1])
plt.xlabel('epochs')
plt.legend(['train', 'test'], loc='upper left')
plt.savefig(output_root+'spleen.png')
```

Figure 3.17 – Illustration of Visualising Results function.

## 3.6 Experimental Results

In this part, we provide the outcomes of our automated segmentation model on several examinations from the utilized dataset, throughout numerous epochs, for the labels Liver, Left Kidney, Right Kidney, and Spleen.

Our results are organized into four sections for three models using two databases. Where, each one describes the demonstration outcomes for a certain organ label (liver, spleen, right Kidney and left Kidney ). The dice and the loss coefficient, as well as the anticipated 2D images of MRI database and 3D images of CT database, were employed as assessment metrics in our experimental findings.

### 3.6.1 The obtained segmentation results of the liver organ

#### 3.6.1.1 3D MRI imaging modality Database

To segment the organ Liver, we have applied the training and testing steps on the exams (1, 2, 3, 5, 8, 10, 13, 15, 19 and 20) for training on T1-DUALin modality, fit into the model and training 250 epochs every epoch over 10 steps for the Unet and Vgg16-Unet models.

The Table 3.3 summarizes all the performances obtained (accuracy and loss) by our model for liver organ segmentation in MRI database. Where the explication of each metric as follows:

- Dice Coef: represent the dice coefficient of training.
- Val Dice: represent the dice coefficient of testing.
- Loss Coef: represent the loss coefficient of training.
- Val Loss: represent the loss coefficient of testing.

The Figure 3.18 and the Figure 3.19 illustrates the obtained curves of dice, validation dice ,loss and validation loss metric for the Unet model and Vgg16-Unet respectively. For the Vgg19-Unet model, we use the exams number (1 and 2) for training set of T1-DUALin modality , the training of model with 100 epochs, 10 as number of steps and using the data augmentation with use the binary cross-entropy as loss metric ( show the Figure 3.20).

Models	Dice Coef	Loss Coef	Val Dice	Val Loss
Unet	0.95	0.04	0.93	0.07
Vgg16-Unet	0.94	0.05	0.91	0.07
Vgg19-Unet	0.93	0.004	0.87	0.01

Table 3.3 – Illustration of the obtained results for liver Organ segmentation in MRI database .

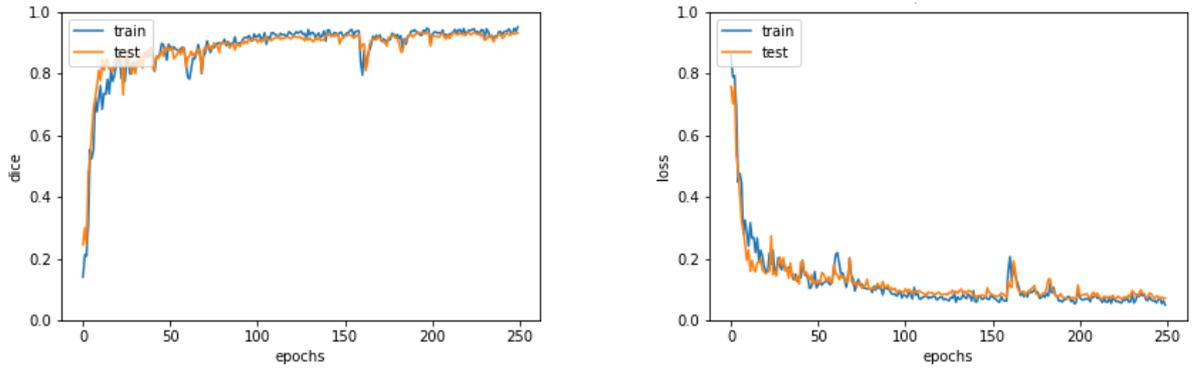


Figure 3.18 – Illustration of dice and loss coefficients of Unet Model for Liver segmentation in MRI modality.

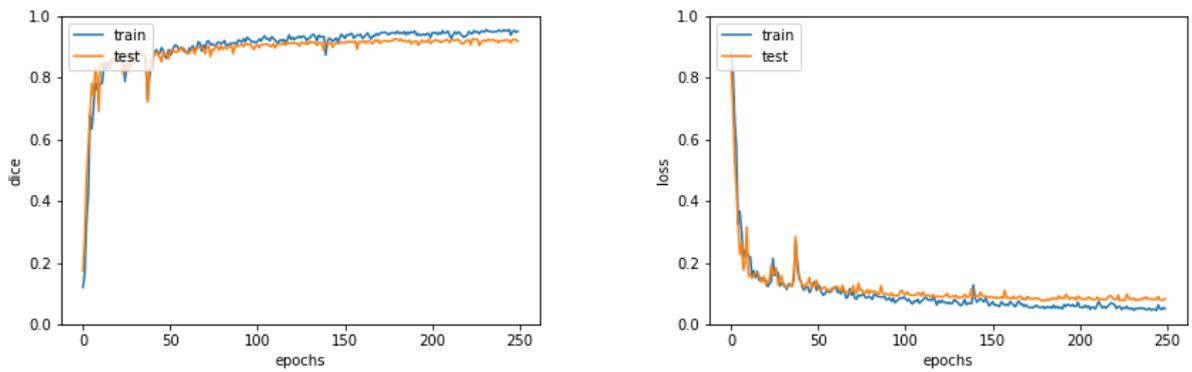


Figure 3.19 – Illustration of dice and loss coefficients of Vgg16-Unet Model for Liver segmentation in MRI modality

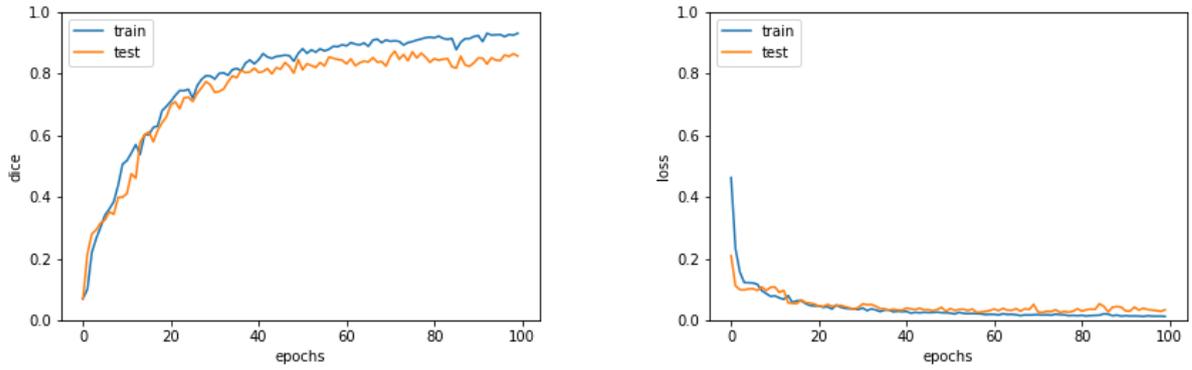


Figure 3.20 – Illustration of dice and loss coefficients of Vgg19-Unet Model for Liver segmentation in MRI modality.

After then, we move to the prediction of our results, we choose two exams id for testing which are 21 and 22. Therefore, we show some slices of these exams for both models (unet and vgg16-unet). The Figure 3.21.(a) represents the original 2D axial slices, Figure 3.21. (b) represents the ground Truth (mask), and Figure 3.21. (c) our predicted results. For Vgg19-Unet model we choose the exam id 5 for predicting our results (see Figure 3.22).

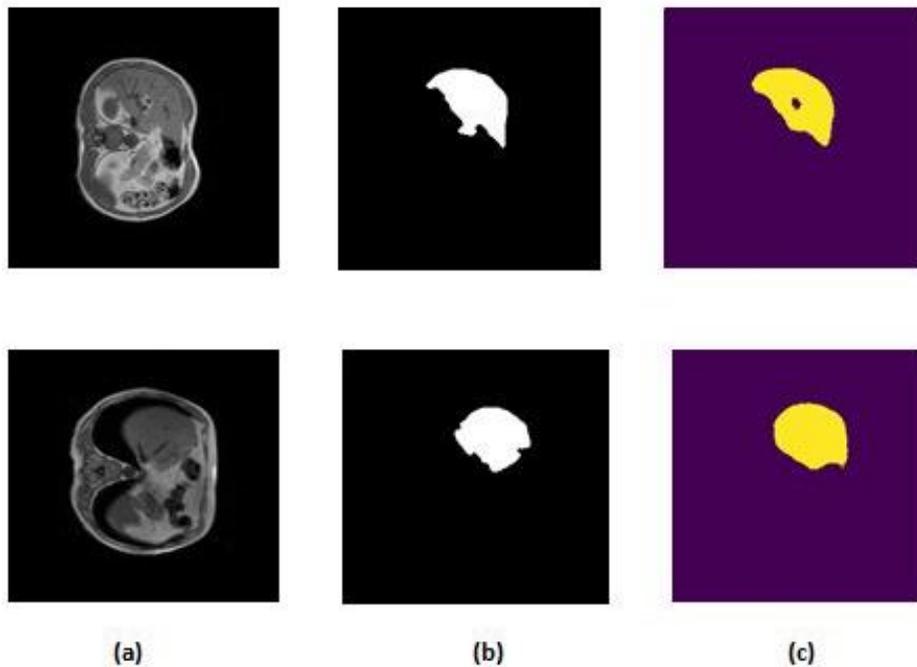


Figure 3.21 – Illustration of the predicted results of Unet, Vgg16-Unet models for Liver segmentation in MRI. a) 2D axial slice, (b) Ground Truth, (c) our predicted results.

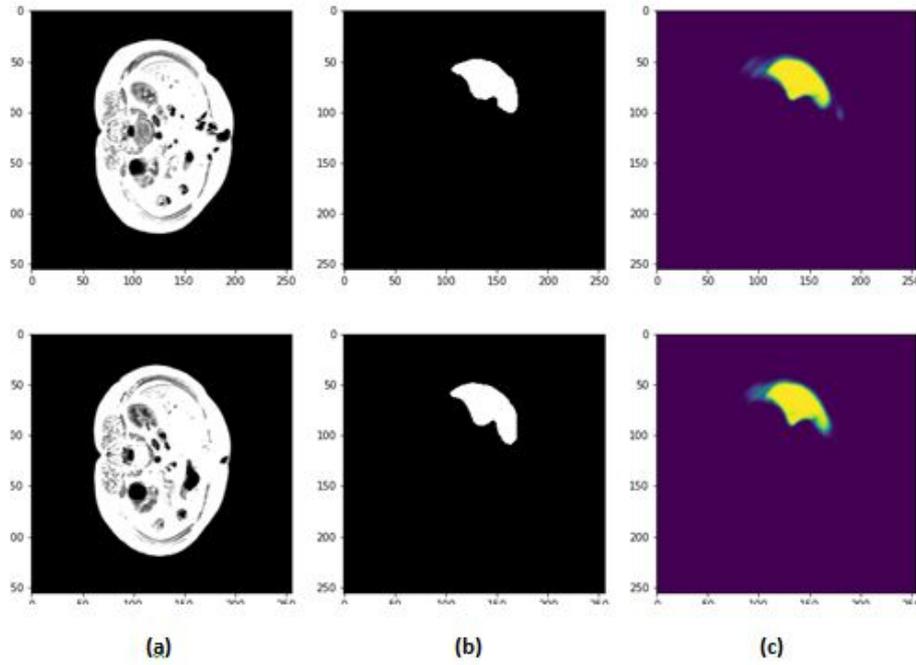


Figure 3.22 – Illustration of the predicted results of Vgg19-Unet model for liver segmentation in MRI. a) 2D axial slice, (b) Ground Truth, (c) our predicted results.

### 3.6.1.2 CT Database

In this part, for segmenting liver organ we use 80% of Data for training Unet model. After, when we have the pre-processed data in 200 epochs and number of batch size is 32. As results we finding for dice coefficient **0.98** , validation dice coefficient **0.96**, for the loss coefficient **0.03**, and validation loss coefficient is **0.01** as showing in the Figure 3.23.

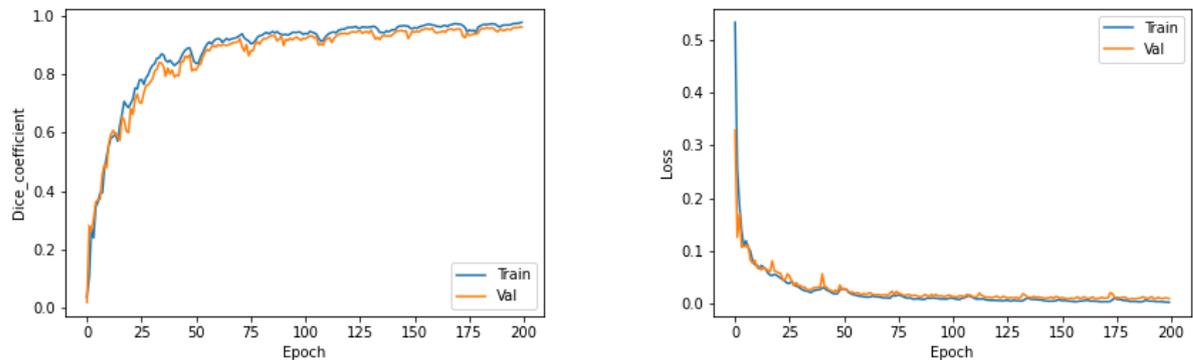


Figure 3.23 – Illustration of the obtained performance (dice and loss coefficients) of Unet Model for Liver organ segmentation in CT images.

For the prediction of our result, we use the rest of data (20%). The Figure 3.24 illustrates the obtained results of prediction.

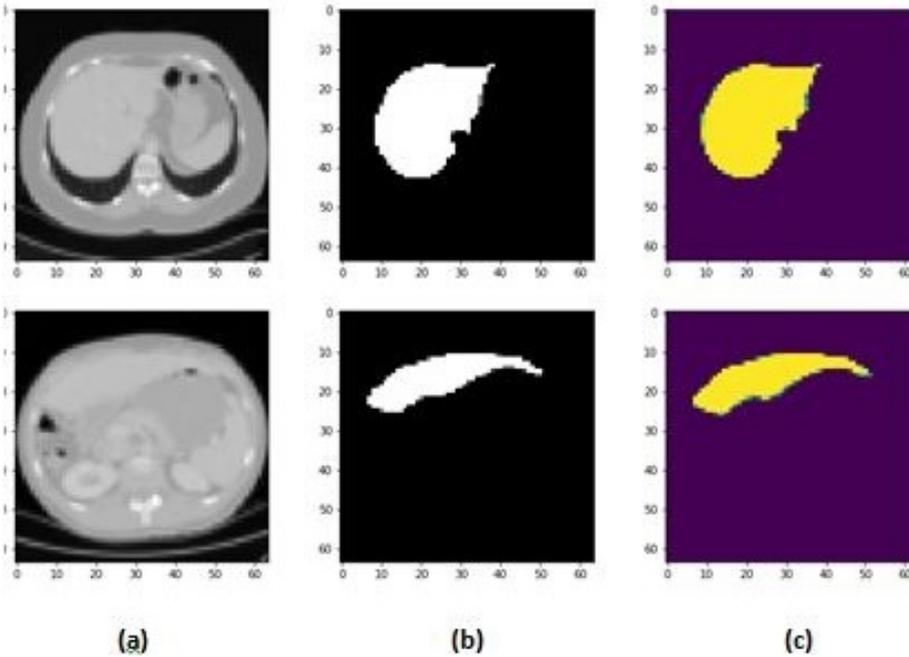


Figure 3.24 – Illustration of the predicted results of Unet models for liver organ segmentation in CT images. a) 3D CT images, (b) Ground Truth, (c) our predicted results.

### 3.6.2 The obtained segmentation results of spleen organ in MRI database

In the case of spleen organ segmentation, for Vgg16-unet model we use the exams ids (1, 2, 3, 5, 8, 10, 13, 15, 19 and 20) for training set of T1-DUALin modality.

Fit into model over 250 epochs and 10 steps with using the data augmentation so the results are shown in the Figure 3.25. Then, we move to Unet model in the training phase we use the exams ids (1,2) of T1-DUALin. We use also augmentation of data for training the model with 100 epochs and 10 steps with used binary crossentropy of loss metric the Figure 3.26 as a results of Unet model. For Vgg19-Unet model we use the same parameters as in Unet model where the obtained results are illustrated in Figure 3.27. The Table 3.4 summarizes all the performances obtained (accuracy and loss) by our model for spleen organ segmentation in MRI database.

Models	Dice Coef	Loss Coef	Val Dice	Val Loss
Vgg16-UNet	0.92	0.07	0.87	0.18
UNet	0.90	0.001	0.87	0.003
Vgg19-UNet	0.89	0.003	0.87	0.01

Table 3.4 – Illustration results of Spleen Organ .

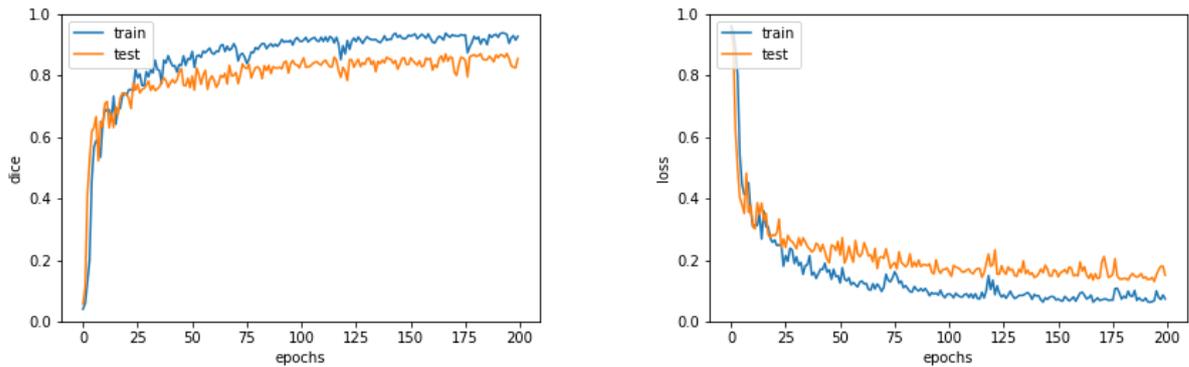


Figure 3.25 – Illustration the dice and loss coefficient of Vgg16-UNet model for Spleen organ in MRI database.

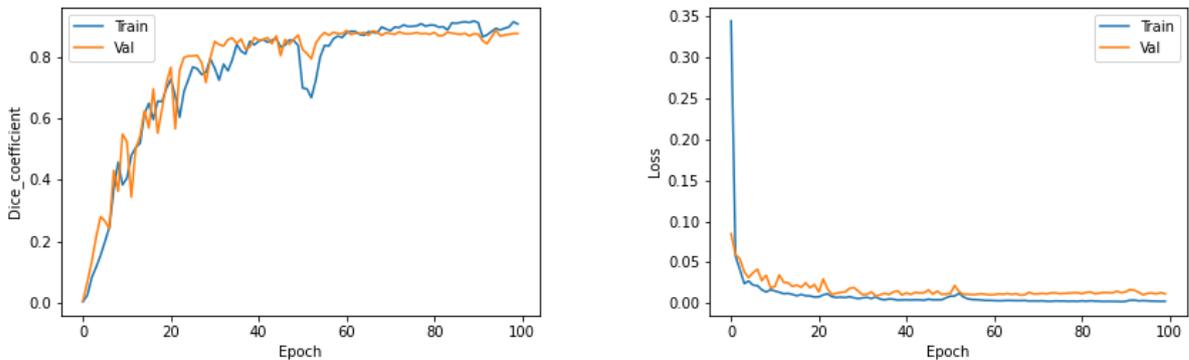


Figure 3.26 – Illustration the dice and loss coefficient of UNet model for Spleen organ in MRI database.

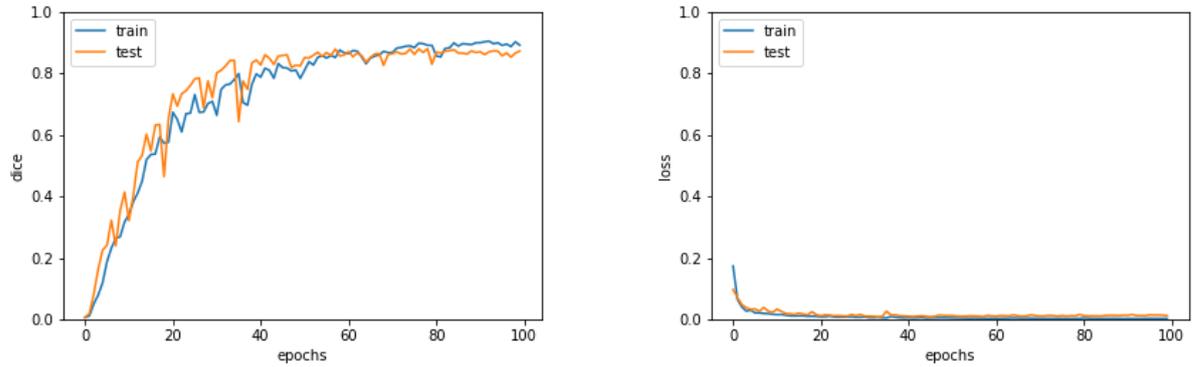


Figure 3.27 – Illustration the dice and loss coefficient of Vgg19-Unet model for spleen organ in MRI database.

For the prediction, use some exams ids of testing set for Vgg16-Unet model we choose exam id 22 with also two 2D slices and for Unet. For Vgg19 models we use the same exam id 5 for predicting. The Figures 3.28, 3.29 and 3.30) illustrated the prediction result for Vgg16-Unet, Unet, and for Vgg16-Unet models respectively.

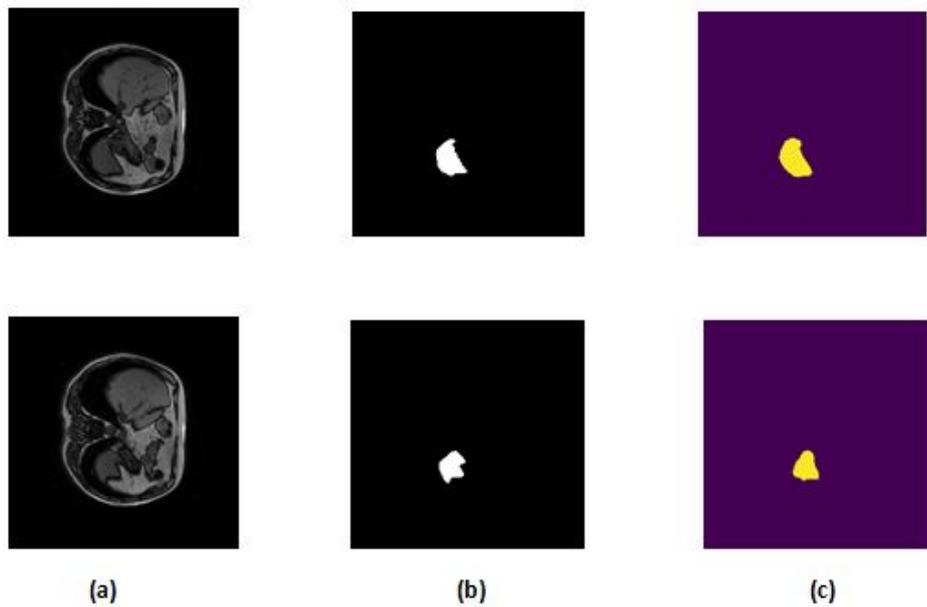


Figure 3.28 – Illustration the the predicted results of Vgg16-Unet model for spleen organ in MRI database. a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

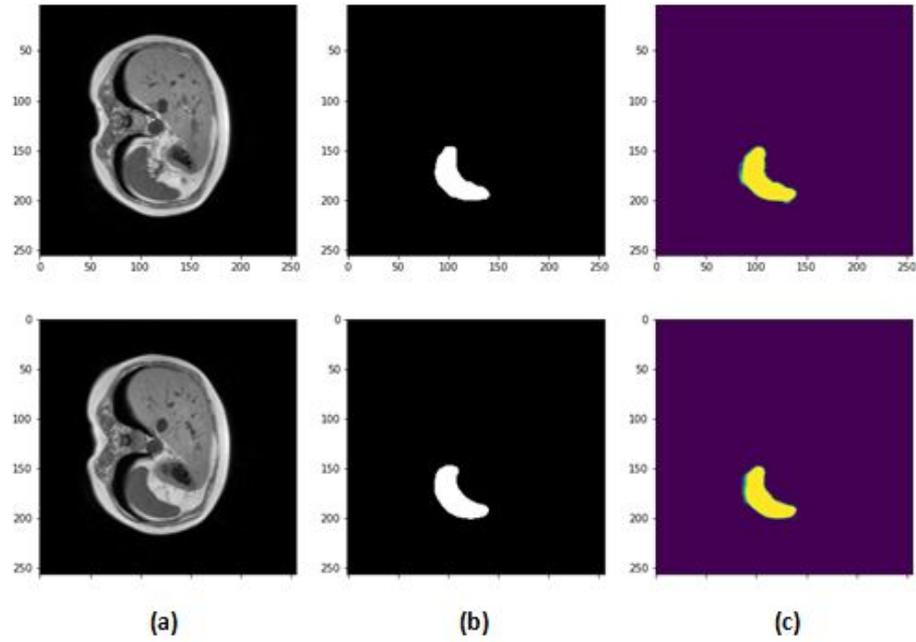


Figure 3.29 – Illustration the the predicted results of Unet model for Spleen organ in MRI database. a) 2D axial slices, (b) Ground Truth, (c) our predicted results.

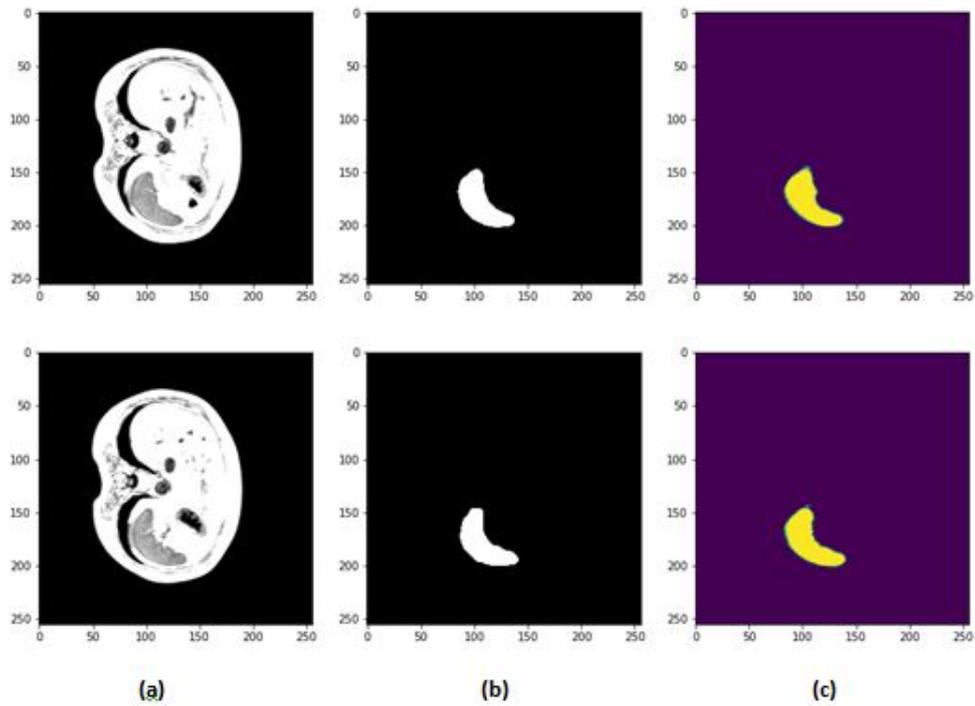


Figure 3.30 – Illustration the the predicted results of Vgg19-Unet model for Spleen organ in MRI database. a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

### 3.6.3 The obtained segmentation results of right kidney organ in MRI database

For the segmentation of right kidney organ with Vgg16-Unet model we use the exams (1, 2, 3, 5, 8, 10, 13, 15, 19 and 20) for training set of T2-SPIR modality. We train the model for 250 epochs and 10 steps by using the data augmentation, the Figure 3.31 represents the plot results of Vgg16-Unet model. For Unet and Vgg19-Unet models we use the same way which is for training set the exams (1 and 2) set also of T2-SPIR modality fit into the model with 100 epochs and 10 of steps by used data augmentation, shown in Figure 3.32, Figure 3.6.3 the plot results of Unet, Vgg19-Unet models . So, we collect the results of those models in the Table 3.5.

Models	Dice Coef	Loss Coef	Val Dice	Val Loss
Vgg16-Unet	0.94	0.07	0.91	0.08
Unet	0.89	0.002	0.86	0.01
Vgg19-Unet	0.88	0.002	0.87	0.006

Table 3.5 – Illustration results of right kidney Organ .

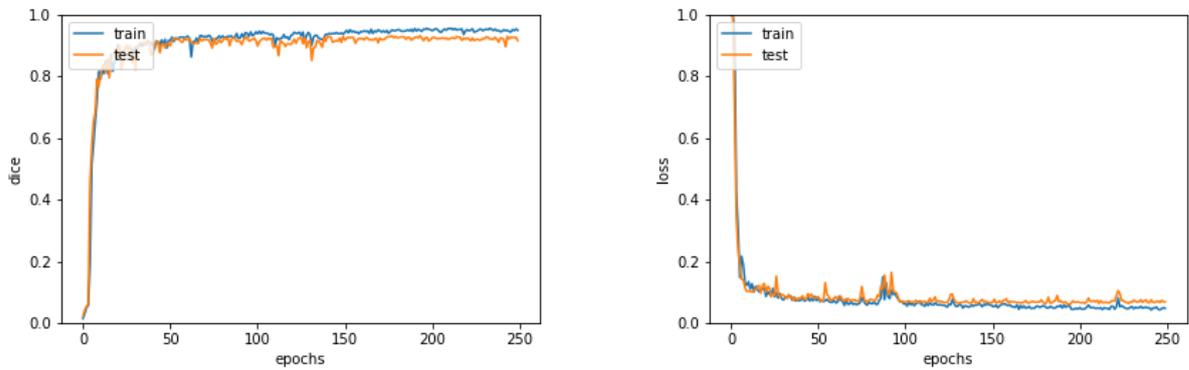


Figure 3.31 – Illustration the dice and loss coefficient of Vgg16-Unet model for Right Kidney organ in MRI database

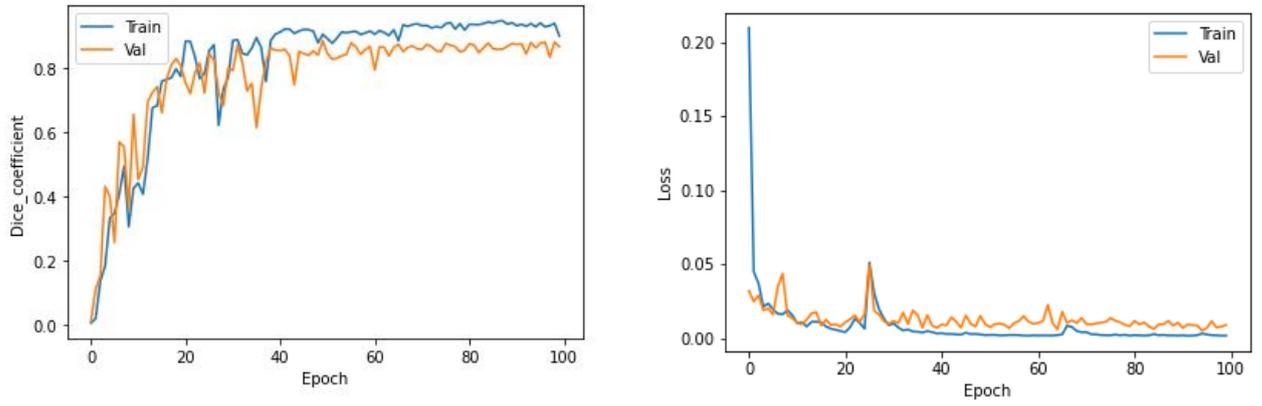


Figure 3.32 – Illustration the dice and loss coefficient of Unet model for Right Kidney organ.

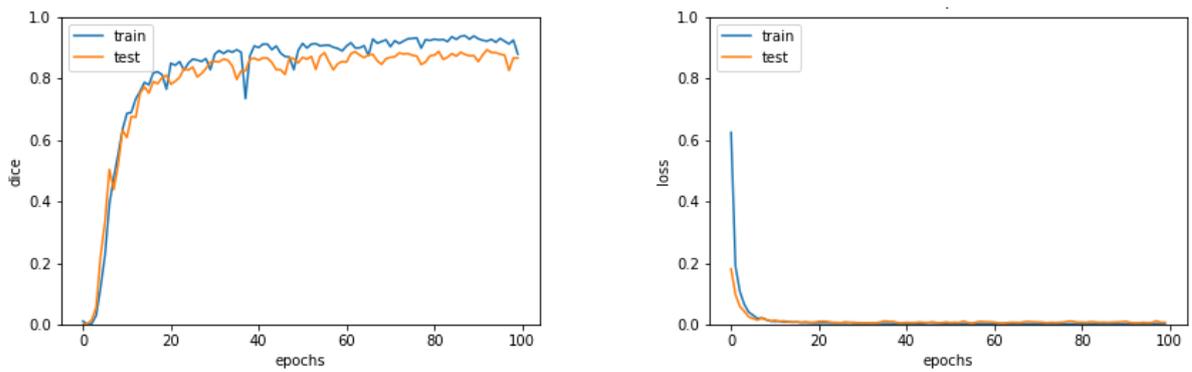


Figure 3.33 – Illustration the dice and loss coefficient of Vgg19-Unet model for Right Kidney organ in MRI database

for the phase of prediction of our results, we follow the same parameters as the previous organ (spleen) as showing in the figures ( 3.34, 3.35 and 3.36).

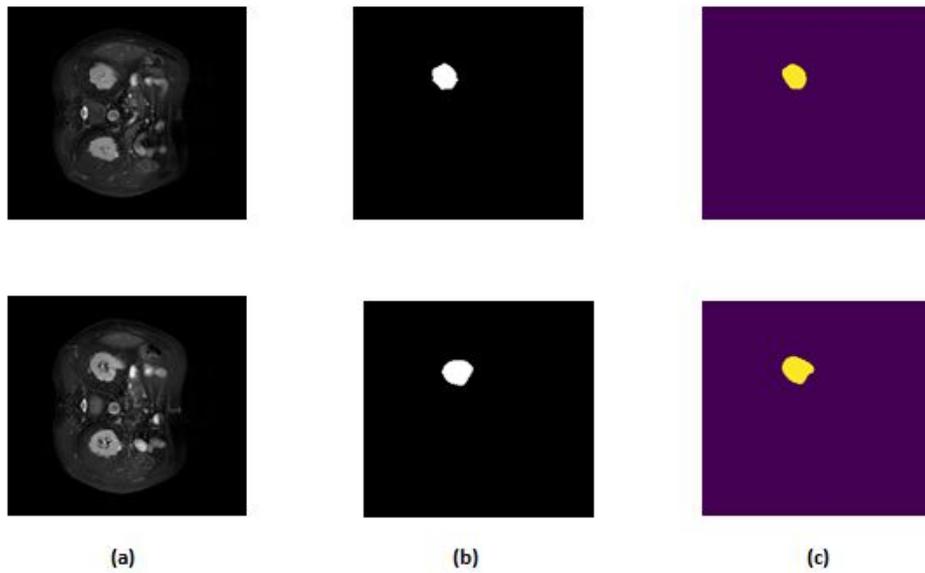


Figure 3.34 – Illustration the the predicted results of Vgg16-Unet model for Right Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

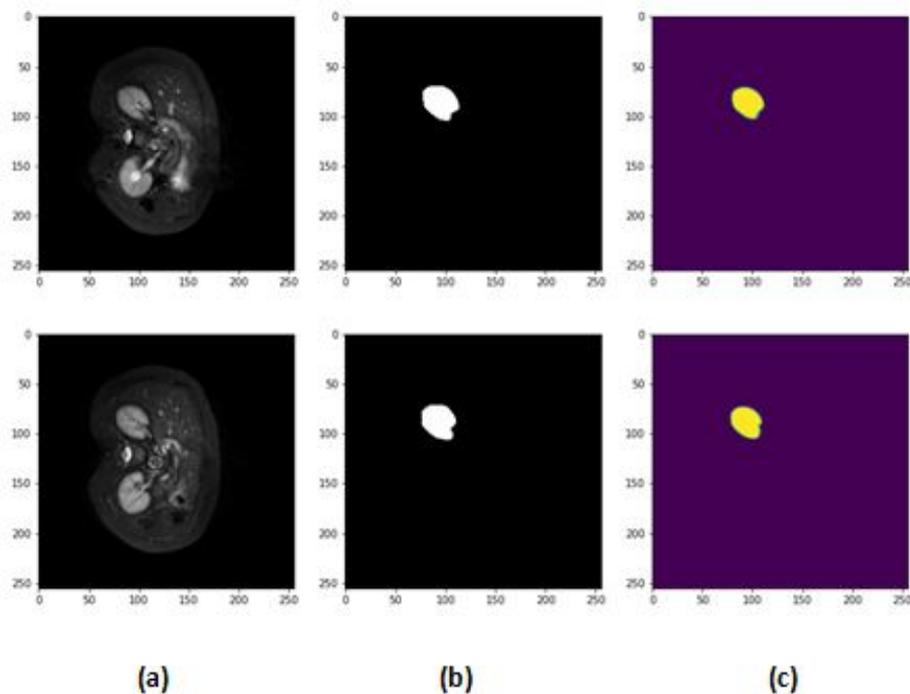


Figure 3.35 – Illustration the the predicted results of Unet model for Right Kidney organ. a) 2D axial slices, (b) Ground Truth, (c) our predicted results.

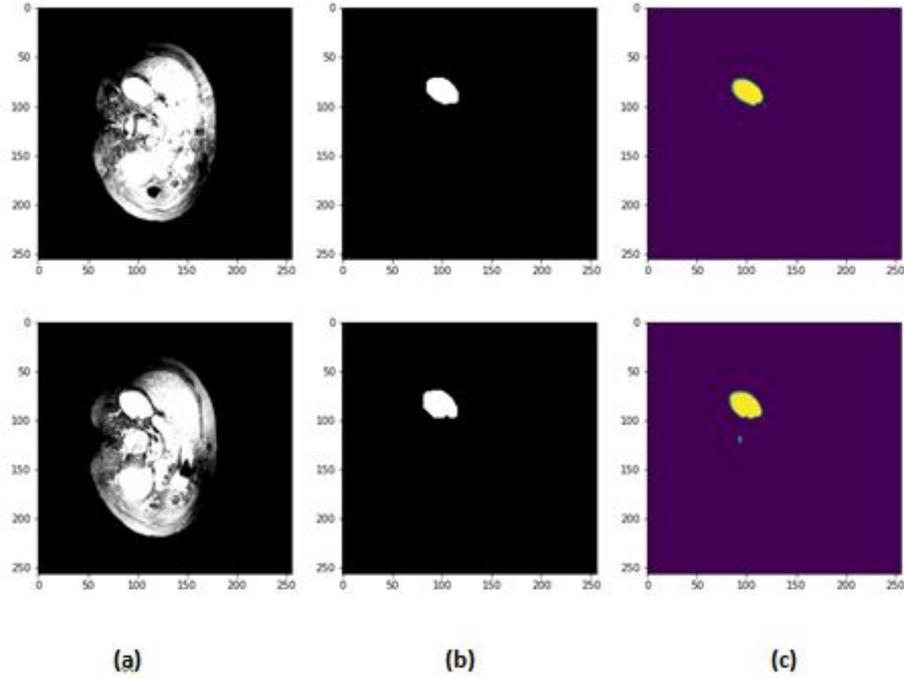


Figure 3.36 – Illustration the the predicted results of Vgg19-Unet model for Right Kidney organ. a) 2D axial slices, (b) Ground Truth, (c) our predicted results.

### 3.6.4 The obtained segmentation results of Left Kidney in MRI database

For the left kidney organ segmentation, we use for the training of Vgg16-Unet model the exams of ids (1, 2, 3, 5, 8, 10, 13, 15, 19 and 20) of T2-SPIR modality. Fit the model into 250 epochs and 10 of number of steps with the data augmentation. For Unet model we train it with the T2-SPIR modality of exams ids (1,2), fit into model with over 50 epochs with 10 steps and also used the data augmentation. The same way for Vgg19-Unet but training with 100 epochs. For represent it results show the Table 3.7 with plotting those results as showing in (Figure 3.37 for Vgg16-Unet , (Figure 3.38 for Unet model and for Vgg19-Unet shown in (Figure 3.39) .

Models	Dice Coef	Loss Coef	Val Dice	Val Loss
Vgg16-Unet	0.95	0.04	0.91	0.07
Unet	0.89	0.003	0.85	0.009
Vgg19-Unet	0.908	0.003	0.86	0.009

Table 3.6 – Illustration results of left kidney Organ .

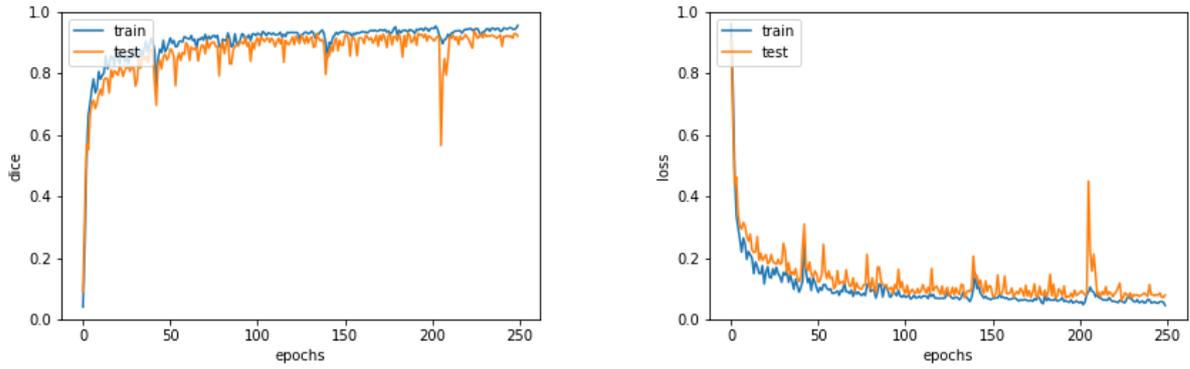


Figure 3.37 – Illustration the dice and loss coefficient of Vgg16-Unet model for Left Kidney organ in MRI.

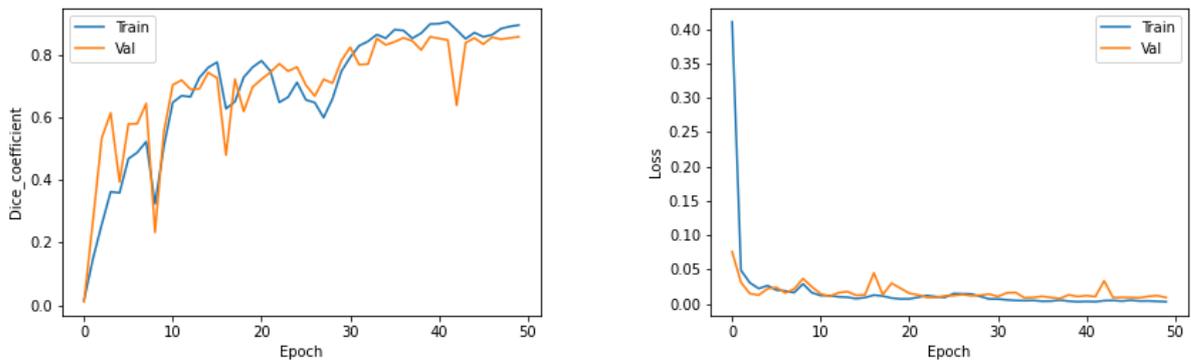


Figure 3.38 – Illustration the dice and loss coefficient of Unet model for Left Kidney organ in MRI.

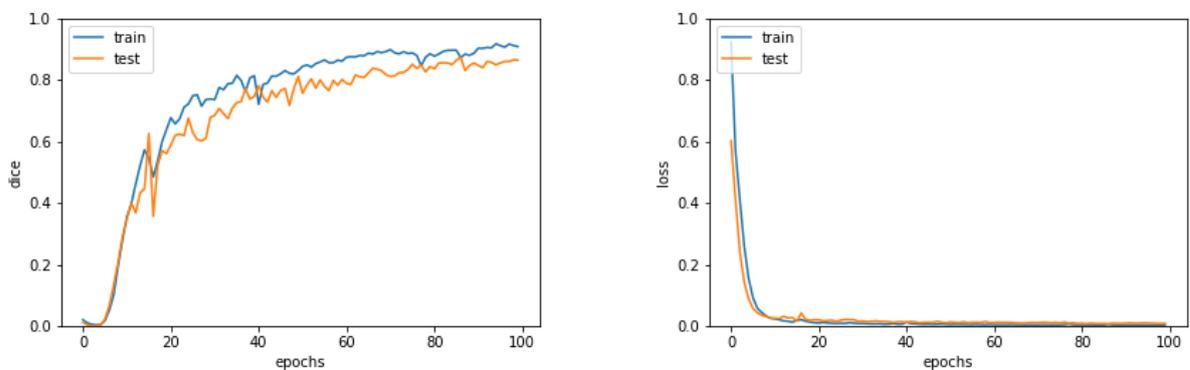


Figure 3.39 – Illustration the dice and loss coefficient of Vgg19-Unet model for Left Kidney organ in MRI.

After then we predict our results by using as testing set the exams ids (21,22). To showing some 2D slices with there masks and there predicted results (see the Figure 3.40) for Vgg16-Unet. For Unet and Vgg19-Unet models we use the exam id 5 for testing and predict our results (see Figure 3.41 and Figure 3.42).

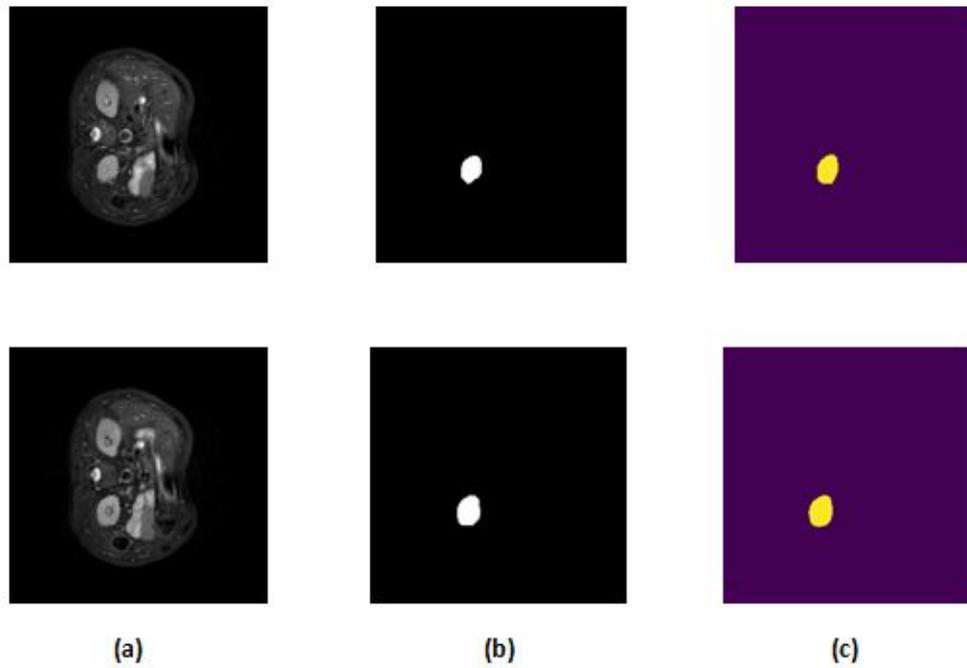


Figure 3.40 – Illustration the the predicted results of Vgg16-Unet model for Left Kidney organ.a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

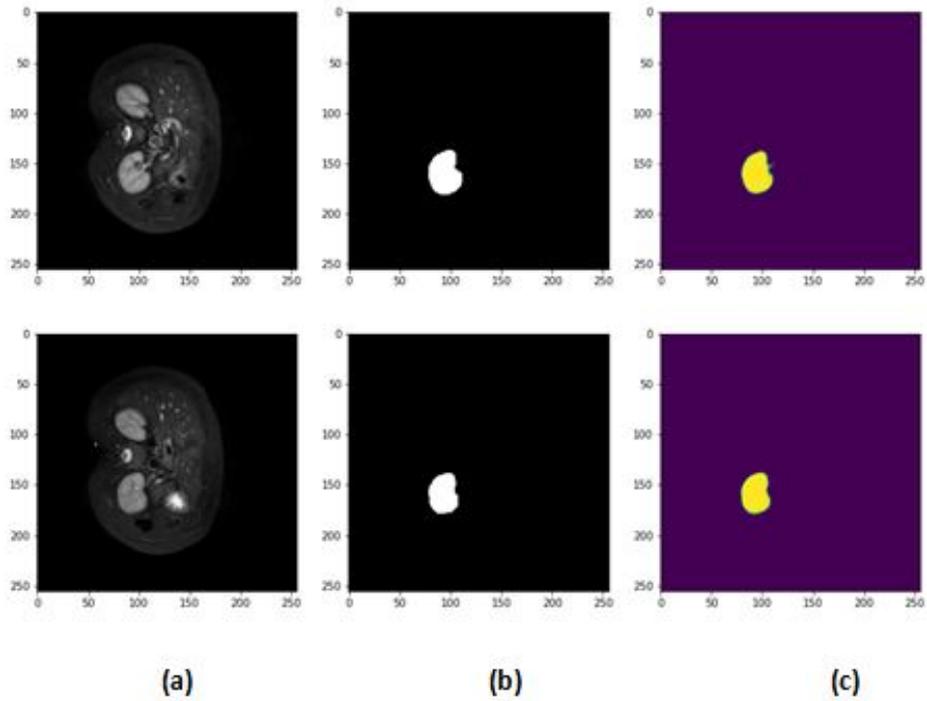


Figure 3.41 – Illustration the the predicted results of Unet model for Left Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

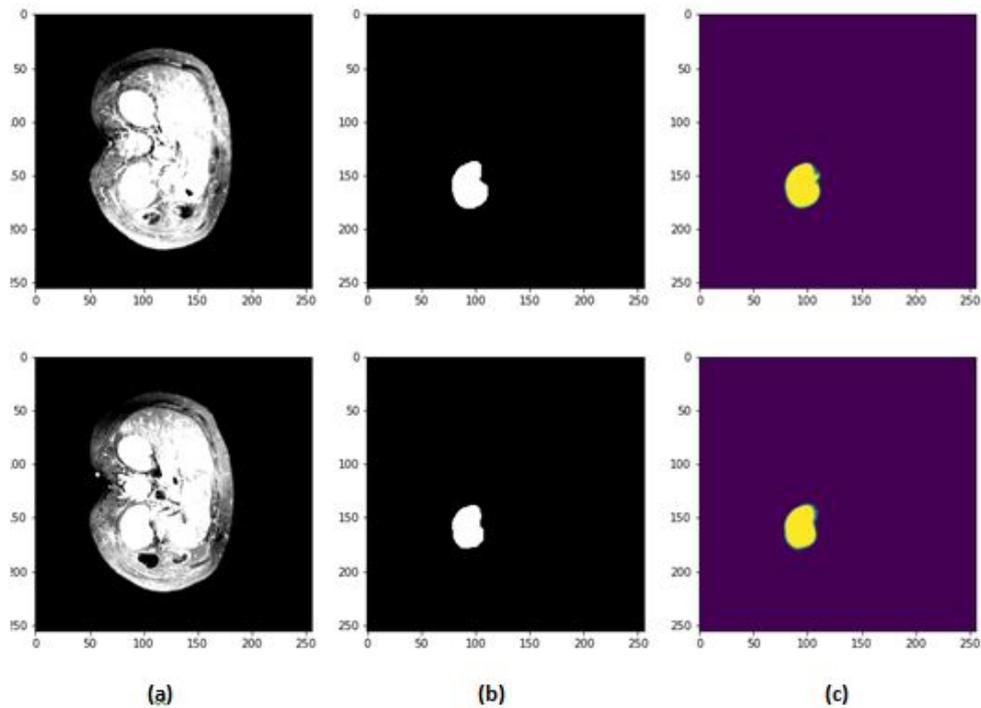


Figure 3.42 – Illustration the the predicted results of Vgg19-Unet model for Left Kidney organ. a) 2D axial slices , (b) Ground Truth, (c) our predicted results.

### 3.6.5 Results Comparison

In order to evaluate our model's performance, we compare our obtained results to some existing deep learning-based approaches in the literature. In general, our approach performs well and provides good results compared to other proposed works in the same context and that are used the same architectures (Unet, Vgg16-Unet, Vgg19-Unet).

In the case of liver organ segmentation, using MRI database we get for *Unet model* a good dice about **0.95**, comparing to **0.85** and **0.90** respectively in [55] and in [54]. and for *Vgg16-Unet model* also we get a good dice about **0.94**, comparing to **0.91** in [54]. For *Vgg19-Unet model* we get the dice coefficient about **0.93**, comparing to **0.92** in [54]. Using CT database we get a excellent dice **0.98**, comparing to the dice of MRI database **0.95** and **0.97** in [54].

For the case of *Spleen* organ, we get for *Unet model* a dice of **0.90** comparing to **0.84** and **0.81** respectively in [55] and in [54]. For *Vgg16-Unet model* the dice is **0.92**, comparing to **0.85** in [54]. And for *Vgg19-Unet model* we get a dice about **0.89**, comparing to **0.82** in [54].

For the *Right Kidney* organ, we get for *Unet model* a dice of **0.89** comparing to **0.86** and **0.81** respectively in [55] and in [54]. For *Vgg16-Unet model* the dice is **0.94**, comparing to **0.91** in [54]. And in *Vgg19-Unet model* we obtain dice for **0.88**, comparing to **0.86** in [54].

About the *Left Kidney* organ, we get for *Unet model* a dice of **0.89** comparing to **0.87** and **0.90** respectively in [55] and in [54]. Using *Vgg16-Unet model* the dice is **0.95**, comparing to **0.91** in [54]. And in *Vgg19-Unet model* we obtain dice for **0.908**, comparing to **0.906** in [54]. The Table 3.7 summarizes all the comparison between our obtained performances and the results of existing approaches.

<i>Organ</i>	<i>Liver</i>	<i>Spleen</i>	<i>Rkidney</i>	<i>Lkidney</i>
<i>MRI Database</i>				
<i>Our Unet</i>	<b>0.95</b>	<b>0.90</b>	<b>0.89</b>	0.89
<i>Unet[54]</i>	0.90	0.81	0.88	0.90
<i>Unet [55]</i>	0.85	0.84	0.86	0.87
<i>Our Vgg16-Unet</i>	<b>0.94</b>	<b>0.92</b>	<b>0.94</b>	<b>0.95</b>
<i>Vgg16-Unet[54]</i>	0.91	0.85	0.91	0.91
<i>Our Vgg19-Unet</i>	<b>0.93</b>	<b>0.89</b>	<b>0.88</b>	<b>0.908</b>
<i>Vgg19-Unet[54]</i>	0.92	0.82	0.86	0.906
<i>CT Database</i>				
<i>Our Unet</i>	<b>0.98</b>	-	-	-
<i>Unet [54]</i>	0.97	-	-	-

Table 3.7 – Comparison of the obtained Dice on each label with some works proposed in literature.

### 3.6.6 Discussion

According to our obtained results, we can see that we have good results with our three models. In particular, one model is better than others in the case of MRI database which is the vgg16-unet. Where, we get excellent results for three classes ( spleen, right kidney and left kidney). For the Unet and vgg19-unet models, we can observe that the unet model gets better results than vgg16-unet for liver organ (CT,MRI), and then vgg19-unet for three classes (liver, spleen and right kidney) , just the left kidney we get good results for vgg19-unet comparing to unet model. Furthermore, the training dice coefficient and the validation dice values are very close which means that the use of data augmentation helps us to eliminate the over-fitting problem cooperated with regularization layer which is Dropout couch specifically for the vgg19-unet model.

## 3.7 Conclusion

This chapter described both the overall and detailed design of our segmentation application. Also provided various tools, packages, and APIs necessary for our implementation. In practise, the observed results across multiple topics indicate highly promising outcomes when compared to existing deep learning-based algorithms for other works.

# General conclusion

The segmentation of medical images is still a vast field of study. In the context of abdominal imaging, the goals of image segmentation include assisting in diagnosis, tracking the evolution of the patient's condition, performing clinical tests...etc.

The goal of our research is to automatically segment human abdominal organs which are liver, spleen, right kidney and left kidney using computed tomography (CT) and also magnetic resonance imaging (IRM) using three deep learning network models Unet, Vgg16-Unet and Vgg19-unet. In fact, they have shown promising results in recent years. The use of a CPU during the training phase causes the execution time to be too long. To solve this problem, deep convolutional neural networks deployed on GPUs rather than CPUs must be used.

To complete our segmentation work, we used the CHAOS [1] challenge data set created in 2019 to learn the models and perform the validation and testing phases. The network parameters are difficult to define in advance. This is why we have defined many models with different architectures in order to achieve the best results in terms of precision and error. Also, we use data augmentation in order to avoid the over fitting problem of data.

The obtained performance generally were satisfying comparing to some existing approaches. Where, a dice coefficient for organs liver, spleen, right kidney, respectively is of (95% (MRI), 98% (CT)), 90%, 89% and 89% for Unet model, 94%, 92%, 94% and 95% for Vgg16-Unet model, 93%, 89%, 88% and 90.8% for Vgg19-Unet model. These results allow us to conclude that the use of semantic segmentation methods (deep learning) allows for better segmentation results.

Several viewpoints can be considered in the extension of this work, including:

- Using other larger data sets for the test.
- Exploiting and combining different conventional network such as CNN network to further improve segmentation accuracy.
- Extending the model to segment all the others organs such as Stomach, Pancreas, Gallbladder, small and large intestines and other.
- Using the CT images for the segmentation of all organs in order to obtained a better accuracy of training and testing.

# Bibliography

- [1] Ali Emre Kavur et al. *CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data*. Version v1.03. Zenodo, Apr. 2019. DOI: [10.5281/zenodo.3362844](https://doi.org/10.5281/zenodo.3362844). URL: <https://doi.org/10.5281/zenodo.3362844>.
- [2] Belladgham Aicha. “Segmentation D’images Et Morphologie Mathematique : Application A L’imagerie Medicale De L’abdomen.” PhD thesis. Université Abou Bekr Belkaid - Tlemcen, 2014.
- [3] Moe Moe Myint and Theingi Myint. “Effective kidney segmentation using gradient based approach in abdominal CT images”. In: *International conference on future computational technologies ICFCT*. 2015, pp. 130–5.
- [4] Thorsten M Buzug. “Computed tomography”. In: *Springer handbook of medical technology*. Springer, 2011, pp. 311–342.
- [5] Raj Acharya et al. “Biomedical imaging modalities: a tutorial”. In: *Computerized Medical Imaging and Graphics* 19.1 (1995), pp. 3–25.
- [6] A.M. Jain. “Survey on Medical Image Processing Techniques - CT an Overview”. In: (Jan. 2014).
- [7] Girish Katti, Syeda Arshiya Ara, and Ayesha Shireen. “Magnetic resonance imaging (MRI)–A review”. In: *International journal of dental clinics* 3.1 (2011), pp. 65–70.
- [8] Nour el houda MAHDJOUB and Youkana Imane. “Deep learning for the segmentation of MRI brain tissues”. In: *Master memory*. University of Biskra, 2020.
- [9] Rohini Paul Joseph, C Senthil Singh, and M Manikandan. “Brain tumor MRI image segmentation and detection in image processing”. In: *International Journal of Research in Engineering and Technology* 3.1 (2014), pp. 1–5.
- [10] Peter Lam. “medicalnewstoday”. In: (July 2018).
- [11] BOUGHEFIR DJIHANE and Youkana Imane. “Abdominal MRI Multi-Organ segmentation using Deep learning”. In: *Master memory*. University of Biskra, 2021.
- [12] Louiza BENHIZIA and Sabrina BENBATATA. “Segmentation of Multiple Sclerosis in MR images by Deep Learning”. In: (2021).

- [13] Cheng Fang et al. “Radiological anatomy of the abdomen”. In: *Surgery (Oxford)* (2021).
- [14] Tim Newman. “Where are the kidneys located, what do they do, and what do they look like?” <https://www.medicalnewstoday.com/articles/305488>. 2021.
- [15] MD Carol DerSarkissian. “Picture of the Spleen”. <https://www.webmd.com/digestive-disorders/picture-of-the-spleen>. 2021.
- [16] Xiangbin Liu et al. “A review of deep-learning-based medical image segmentation methods”. In: *Sustainability* 13.3 (2021), p. 1224.
- [17] MS Abirami and Dr T Sheela. “Analysis of image segmentation techniques for medical images”. In: *Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications*. 2014.
- [18] Nikita Sharma, Mahendra Mishra, and Manish Shrivastava. “Colour image segmentation techniques and issues: an approach”. In: *International Journal of Scientific & Technology Research* 1.4 (2012), pp. 9–12.
- [19] Nida M Zaitoun and Musbah J Aqel. “Survey on image segmentation techniques”. In: *Procedia Computer Science* 65 (2015), pp. 797–806.
- [20] Donald Michie, David J Spiegelhalter, and Charles C Taylor. “Machine learning, neural and statistical classification”. In: (1994).
- [21] Taiwo Oladipupo Ayodele. “Types of machine learning algorithms”. In: *New advances in machine learning* 3 (2010), pp. 19–48.
- [22] Youguo Li and Haiyan Wu. “A clustering method based on K-means algorithm”. In: *Physics Procedia* 25 (2012), pp. 1104–1109.
- [23] Xiaojin Jerry Zhu. “Semi-supervised learning literature survey”. In: (2005).
- [24] Zhiyuan Chen and Bing Liu. “Lifelong machine learning”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12.3 (2018), pp. 1–207.
- [25] Shaveta Dargan et al. “A survey of deep learning and its applications: a new paradigm to machine learning”. In: *Archives of Computational Methods in Engineering* 27.4 (2020), pp. 1071–1092.
- [26] Xueyuan Zhou and Mikhail Belkin. “Semi-supervised learning”. In: *Academic press Library in signal processing*. Vol. 1. Elsevier, 2014, pp. 1239–1269.
- [27] Batta Mahesh. “Machine learning algorithms-a review”. In: *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), pp. 381–386.
- [28] Richard S Sutton. “Introduction: The challenge of reinforcement learning”. In: *Reinforcement Learning*. Springer, 1992, pp. 1–3.

- [29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [30] Haonan Xiao et al. “A review of deep learning-based three-dimensional medical image registration methods”. In: *Quantitative Imaging in Medicine and Surgery* 11.12 (2021), p. 4895.
- [31] Shaveta Dargan et al. “A survey of deep learning and its applications: A new paradigm to machine learning”. In: *Archives of Computational Methods in Engineering* (2019), pp. 1–22.
- [32] Patrick Ferdinand Christ et al. “Automatic liver and tumor segmentation of CT and MRI volumes using cascaded fully convolutional neural networks”. In: *arXiv preprint arXiv:1702.05970* (2017).
- [33] Andrej Krenker, Janez Bešter, and Andrej Kos. “Introduction to the artificial neural networks”. In: *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech* (2011), pp. 1–18.
- [34] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. “Understanding of a convolutional neural network”. In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6.
- [35] BEN CHIKH MASSOUDA. “Algorithmes d’Alignement d’Image à Base de Réseaux de Neurones Convolutionnels (CNN)”. PhD thesis. Faculté des Sciences et Technologies, 2021.
- [36] Rikiya Yamashita et al. “Convolutional neural networks: an overview and application in radiology”. In: *Insights into imaging* 9.4 (2018), pp. 611–629.
- [37] Weihong Xu et al. “Efficient fast convolution architectures for convolutional neural network”. In: *2017 IEEE 12th International Conference on ASIC (ASICON)*. IEEE. 2017, pp. 904–907.
- [38] Lichao Wu and Guilherme Perin. “On the Importance of Pooling Layer Tuning for Profiling Side-Channel Analysis”. In: *International Conference on Applied Cryptography and Network Security*. Springer. 2021, pp. 114–132.
- [39] Camila Alves Dias et al. “Using the Choquet integral in the pooling layer in deep learning networks”. In: *North american fuzzy information processing society annual conference*. Springer. 2018, pp. 144–154.
- [40] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. “Activation functions in neural networks”. In: *towards data science* 6.12 (2017), pp. 310–316.
- [41] Artem Sevastopolsky. “Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network”. In: *Pattern Recognition and Image Analysis* 27.3 (2017), pp. 618–624.

- [42] Zhitao Xiao et al. “Segmentation of lung nodules using improved 3D-UNet neural network”. In: *Symmetry* 12.11 (2020), p. 1787.
- [43] Md Zahangir Alom et al. “Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation”. In: *arXiv preprint arXiv:1802.06955* (2018).
- [44] Md. Zahangir Alom et al. “A State-of-the-Art Survey on Deep Learning Theory and Architectures”. In: *Electronics* 8 (Mar. 2019), p. 292. DOI: [10.3390/electronics8030292](https://doi.org/10.3390/electronics8030292).
- [45] Fabian Isensee et al. “nnu-net: Self-adapting framework for u-net-based medical image segmentation”. In: *arXiv preprint arXiv:1809.10486* (2018).
- [46] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.
- [47] Hercules Dalianis. “Evaluation metrics and evaluation”. In: *Clinical text mining*. Springer, 2018, pp. 45–53.
- [48] Kelly H Zou et al. “Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports”. In: *Academic radiology* 11.2 (2004), pp. 178–189.
- [49] Abdel Aziz Taha and Allan Hanbury. “Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool”. In: *BMC medical imaging* 15.1 (2015), pp. 1–28.
- [50] Christophe Pere. “What are Loss Functions?” <https://towardsdatascience.com/what-is-loss-function-1e2605aeb904>. 2020.
- [51] Meg F Bobo et al. “Fully convolutional neural networks improve abdominal organ segmentation”. In: *Medical Imaging 2018: Image Processing*. Vol. 10574. SPIE. 2018, pp. 750–757.
- [52] Alexander D Weston et al. “Automated abdominal segmentation of CT scans for body composition analysis using deep learning”. In: *Radiology* 290.3 (2019), pp. 669–679.
- [53] Yuhua Chen et al. “Fully automated multiorgan segmentation in abdominal magnetic resonance imaging with deep neural networks”. In: *Medical physics* 47.10 (2020), pp. 4971–4982.
- [54] Pierre-Henri Conze et al. “Abdominal multi-organ segmentation with cascaded convolutional and adversarial deep networks”. In: *Artificial Intelligence in Medicine* 117 (2021), p. 102109.
- [55] Pedro Furtado. “Improving Deep Segmentation of Abdominal Organs MRI by Post-Processing”. In: *BioMedInformatics* 1.3 (2021), pp. 88–105.

- [56] Giang Nguyen et al. “Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey”. In: *Artificial Intelligence Review* 52.1 (2019), pp. 77–124.
- [57] Anaconda software group. *Anaconda Software Distribution. Computer software*. <https://anaconda.com>. 2016.
- [58] S Team. “Spyder Website”. In: *Spyder-ide. org. Recuperado el 21* (2019).
- [59] Wolfgang Beer. *Applied Artificial Intelligence: Neural networks and deep learning with Python and TensorFlow*. Wolfgang Beer.
- [60] The Scipy community. *Numpy Manual*. Numpy organization. Jan. 2020.
- [61] Introducere CONCEPTE ȘI NOȚIUNI GENERALE. “<http://opencv.org>”. In: ().
- [62] John Hunter, Darren Dale, and Eric Firing. “matplotlib: Python plotting”. In: *Matplotlib. org* (2012).
- [63] Bharath K. “*U-Net Architecture For Image Segmentation*”. <https://blog.paperspace.com/unet-architecture-image-segmentation/>. 2022.
- [64] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).