People's Democratic Republic of Algeria
Ministery of Higher Education and Scientific Research
Mohamed El Bachir El Ibrahimi University of Bordj Bou Arréridj
Faculty of Mathematics and Informatics
Informatics Department

DISSERTATION

Presented in fulfillment of the requirements of obtaining the degree

Master in Informatics

Specialty: Networks and Multimedia

THEME

Textured Image Segmentation Using Deep LawsNet.

Presented by :

Benchikh Ikram

Kouadria Houda

Publicly defended on: 19/06/2024

In front of the jury composed of:

President: Mostefai Messaoud

Examiner: Nouioua Farid

Supervisor: Foudhil Belhadj

2023/2024

Dedication

We dedicate this modest work

To those whom we love very much and who have supported us throughout.

Throughout the project

To our parents, may God Promotes good health and longevity.

And of course

Our brothers who love us, without forgetting the sisters.

The family of Benchikh Ikram, Kouadria Houda

To our supervisor, Dr. Belhadj Foudhil,

And to all my fellow computer science students

To our dear friends

And to all those who contributed directly or indirectly,

To make this project possible, we hope that this work will be a source of pride and happiness for you.

Acknowledgment

I would like to thank our supervisor,

Mr. Belhadj Foudhil, for his guidance, dedication, availability,

valuable advice, and assistance throughout the development of this work.

We would like to reserve a special place here to thank all

those

who have supported us in one way or another. Helped

and encouraged in the achievement of this modest work.

Abstract

In recent years, image analysis has become crucial across various fields, from medical diagnostics to autonomous vehicles. At the core of image analysis lies segmentation, a critical step that influences the performance of subsequent processes such as object detection and recognition. This study focuses on heart tumor segmentation in CT images using deep learning, particularly convolutional neural networks (CNNs). We utilized the U-Net architecture, renowned for its precision in medical image segmentation, and introduced a modified version called Law-Net, which incorporates Laws filters.Our work is divided into four chapters: an introduction to medical imaging and segmentation techniques, a detailed overview of U-Net architecture, an exploration of Laws filters and the modified U-Net, and an experimental analysis of our model's performance. By applying various segmentation evaluation metrics, we demonstrated the effectiveness of our approach in accurately detecting and segmenting heart tumors.The results indicate that our model, leveraging widely available data and optimized parameters, achieved high performance across all evaluation metrics. This confirms the potential of deep learning in aiding objective medical diagnoses and underscores its capacity to enhance the accuracy and efficiency of tumor detection in CT images. Our deep learning model holds promise for significantly improving the diagnostic process for medical professionals.

Keywords: deep learning,artificial intelligence,convolutional neural network, image segmentation , Unet model , textured image analysis , medical image segmentation,tumor detection .

Table of contents

Dedication..……………………………...

Acknowledgment..i

Abstract...v

Table of contents..v

List of figures...viii

List of tables..x

General Introduction..1

Chapter 01: Medical Image Segmentation..3

1.1. Definition of Medical Imaging..3
1.2. Importance of Medical Imaging in Healthcare...3
1.3. Types of Medical Imaging...3
1.3.1 X-ray Imaging...4
1.3.2 Computed Tomography (CT) OR Computed Axial Tomography (CAT) SCAN….5
1.3.3 Magnetic Resonance Imaging (MRI)..6
1.3.4 Ultrasound...6
1.4. What is Medical Image Segmentation..7
1.5. Methods Used in Medical Image Segmentation...7
1.5.1 Threshold-based Segmentation..8
1.5.2 Region-based Segmentation..8
1.5.3 Edge-based Segmentation...8
1.5.4 Machine Learning-based Segmentation...9
1.6. The Benefits of Medical Image Segmentation..9
1.7. Challenges and Future Directions...9

1.8. Using U-Net....10

Chapter 02 : U-net image segmentation....**11**

2.1. Introduction....11
2.2. U-Net Architecture....11
2.3. Medical Image Segmentation Based On U-net....14
2.3.1 U-net segmentation for tumor segmentation....14
2.4. Challenges....16
2.5. Conclusion....17

Chapter 03 : Modified U-net (Law's-Net)....**18**

3.1. Introduction....18
3.2. Law's masks....18
3.3. Laws Filters Application....19
3.4. Modified U-Net Architecture....23
3.5. Conclusion....23

Chapter 04 : Implementation and results....**25**

4.1. Introduction....25
4.2.1. Programming Language Overview "Python"....25
4.2.2. Programming Language Overview " GOOGLE COLAB "....25
4.3. Test approach....28
4.3.1. Method....28
4.3.2.Data preparation....29
4.3.2.1. Data augmentation....29
4.3.3. Creation of Laws masks....31
4.3.4. Training....34
4.3.5. Evaluation....36
4.3.6. Test predictions....40
4.3.7. Result discussion....40
4.4. Conclusion....40

General Conclusion....**42**

References..**443**

List of figures

Figure 1. An x-ray image taken of a human wrist bone..4

Figure 2. Medical Image Segmentation...6

Figure 3. U-net architecture..12

Figure 4. Examples of U-net applications...13

Figure 5. Results of brain tumor MRI image segmentation using U-net.................................15

Figure 6. Four of Laws most successful masks..18

Figure 7.1. Filtered_image_E5L5...20

Figure 7.2. Filtered_image_E5S5...21

Figure 7.3. Filtered_image_L5S5...21

Figure 7.4. Filtered_image_R5R5...22

Figure 8. Modified U-Net Architecture (Mask production)..23

Figure 9.1. Google Colab..26

Figure 9.2. Google Colab..27

Figure 10. Implementation process...28

Figure 11.1. Original image and its corresponding mask...29

Figure 11.2. vertical flip of brain image..30

Figure 11.3. Horizontal flip of brain image...30

Figure 11.4. Rotation of brain image..31

Figure 12. Creation of Laws masks..31

Figure 13. Filtered Images..33

Figure 16. Some Results of Evaluation..37

Figure 17. Some Output maps after segmentation applied on test data..................................38

List of tables

Table 1. Results of Laws filters Application..20

Table 2. Training performance criteria for two epochs..35

Table 3. The segmentation performances..41

General Introduction

During the last years, image analysis and interpretation have become widely employed in a variety of fields, ranging from diagnostic assistance in medicine to autonomous vehicle navigation, including face recognition, fingerprints, and quality control of manufactured products. In an image analysis system, segmentation is considered the most critical phase because all other processes, such as primitive extraction, object location detection, and object recognition, depend heavily on the quality of segmentation.

We introduce the specific problem of image segmentation to enable a more precise analysis. The process of image segmentation involves the automatic division of an image into distinct regions based on the classification of pixels. This computer vision technique finds utility in various fields, with medical imaging being one of its notable applications.

As mentioned before, medical imaging is one of the areas that uses image segmentation. This technique allows images to be divided into zones for cell counting and detection of changes or anomalies. The analysis of images enables the diagnosis of diseases like cancer, bone age calculations, and scoring embryos in in vitro fertilization procedures. It is also widely used for analyzing satellite images. Segmentation models can detect roads, buildings, or fields.

Image segmentation is increasingly being integrated into on-board cameras in self-driving cars. The models used in this context are quite general and can detect almost all objects that a car could encounter on its path, such as signs and pedestrians. Therefore, we have a need for segmentations that are both accurate and consistent, and it is essential that we provide a method for measuring the reliability of a segmentation technique. Segmentation is a challenging task for several reasons. Firstly, the appearance of the structures of interest can vary greatly between different types of scans and from patient to patient. They are often poorly defined and have very heterogeneous intensities. As a result, it is difficult to design a method that will work well in all cases, and it is often necessary to utilize some prior information about the shape and appearance of the structure. Secondly, medical images are often very large, which can make computation prohibitively expensive. This is particularly the case now that we have 4-D data from time sequences of 3-D scans. Thirdly, and most importantly, the extracted information will most often be used to make very important clinical decisions.

In our project, we will utilize Deep Learning based on one of the most widely used convolutional neural networks, U-Net. This artificial intelligence method aims to enable machines to perceive and comprehend visual content in a manner similar to that of humans. In this field, U-Net has established itself as a remarkable architecture. One of the main advantages of U-Net lies in its ability to accurately segment images, allowing the prediction of individual pixels. To assess the effectiveness of image segmentation, a database consisting of 8104 images and masks . The goal is to compare the segmentation results obtained with modified U-Net to determine if there is an improvement.

To accomplish this, we have organized our thesis into four chapters:

In the first chapter, we start with a general introduction and introduce the fundamental concepts of medical imaging and image segmentation methods.

- The second chapter is dedicated to the architecture of U-Net.

- The third chapter is devoted to Laws filters and modified U-Net.

- In the fourth chapter, we will present the experimental part of our work and discuss the various results obtained.

Then we conclude with a general conclusion.

Chapter 01: Medical Image Segmentation

1.1. Definition of Medical Imaging :

Medical imaging is a specialized field that involves capturing images of the human body and interpreting them for diagnostic, therapeutic (interventional imaging), or monitoring the progression of diseases.

The discipline of "radiology and medical imaging" at the university and hospital discipline “radiology and medical imaging” includes two distinct medical specialties: radiology and medical imaging, and nuclear medicine. Radiology utilizes X-rays, ultrasound, magnetic resonance imaging (MRI), and nuclear medicine with radioactive isotopes. While every modality has specific applications and obstacles of its own, when combined it makes for a very powerful toolbox.

1.2. Importance of Medical Imaging in Healthcare:

Medical imaging plays a crucial role in modern healthcare. The objective is not only to diagnose diseases, monitor their development, and discover how they work but also to improve their treatment. Techniques are being developed to locate sources of infection, target them, and activate the active ingredients of drugs only at the desired location. To destroy well-localized cells using shear waves emitted by an ultrasound machine without the need for surgery. The development of MRI for research also opens up the prospect of a more detailed understanding of this highly complex organ.

1.3. Types of Medical Imaging :

Medical imaging is considered as a part of biological imaging, which has been developed from 19th century onwards. A brief overview of medical imaging is as follows:

In 1895 Roentgen accidentally discovered X-rays. Conventional radiography has been the most widespread medical imaging technique ever science. From 1896 radio-nuclides were for therapy and for metabolic tracer studies rather than imaging. Then γ- ray imaging rectilinear scanner was invented. During World War 2 Sonar Technology and in 1970's ultrasound became widely available in medicine. In 20th century the mathematical principles behind tomographic reconstruction have been understood and positron emission tomography (PET) and X-ray computed tomography (CT) have been developed. Nuclear magnetic resonance has been using for imaging in magnetic resonance imaging (MRI). In 21st century X-rays, MRI, ultrasound kept dominating but more interesting techniques especially imaging is getting included with microscopic as well as macroscopic biological structures (thermal imaging, electrical impedance tomography, scanned probe techniques etc).

1.3.1. X-ray Imaging:

Radiography is a useful imaging technique used in radiology to produce detailed 2-dimensional images of the inside of the body using x-rays. It is used to diagnose various medical conditions and is performed by a radiologist in a hospital; radiography uses X-rays. By extension, the term "radiography" also refers to the x-ray image.

Although other, more advanced imaging techniques are increasingly used in medical practice, radiography retains its crucial role due to its simplicity, speed and relatively low cost, making it a method valuable in many clinical situations. Typically used used for assessment: Broken bones, Arthritis in the joints, Blood vessels, Dental issues, Lungs, Mammography, Foreign objects (such as something swallowed by a child) [1].

Figure 1. An x-ray image taken of a human wrist bone.

1.3.2. Computed Tomography (CT) OR Computed Axial Tomography (CAT) SCAN:

Computed axial tomography (CT) maps X-ray attenuation characteristics of tissue, which are determined by the elemental composition (electron density) of the tissue through which Xrays pass. Using X-ray measurements from a large number of projection view angles, crosssectional images (tomograms) are reconstructed, providing a 2D or 3D map of pixels that are given a numerical value (Hounsfield unit) where 0 is assigned to water and -1000 is assigned to air; lower numbers represent lower electron density. This early detection has been helpful in monitoring target organ injury and in turn developing novel therapeutic target to alleviate and avert them. In particular, using CT to assess metabolic syndrome reported that accumulation of VAT is the best predictor of metabolic syndrome in women and a good predictor of metabolic syndrome in man [2].

1.3.3. Magnetic Resonance Imaging (MRI) :

Magnetic Resonance Imaging is based on advanced technology that causes and detects changes in the direction of the rotating axis of protons present in the water that composes biological tissues.

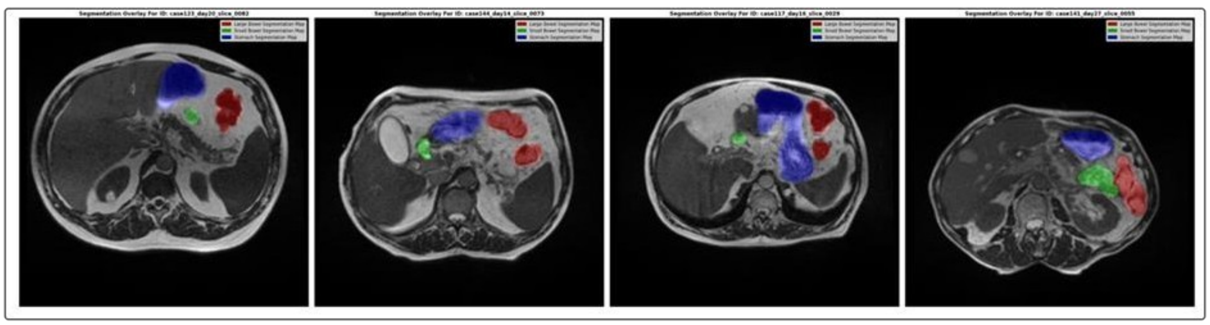
MRI scanners are very useful for imaging the body's non-bony or soft tissues. They vary from computed tomography (CT) in that they do not emit the damaging ionizing radiation of x-rays. MRI provides a considerably clearer view of the brain, spinal cord, and nerves, as well as muscles, ligaments, and tendons, than normal x-rays and CT; for this reason, MRI is often used to image knee and shoulder injuries. Typically used to evaluate: Tendon and ligament tears, Spinal conditions and injuries, Bones and joints, Imaging organs, soft tissue, and other internal structures, Blood vessels [3].

1.3.4. Ultrasound :

Ultrasound (US) is probably the most convenient imaging method Develop methods to quantify tissue mass and types, due to wide availability in clinics Exercise (for diagnostic purposes) and response to treatment, portability and relatively low cost. Ultrasonic transducers generate sound waves that are reflected by tissue, producing converted echoes Converted into signals for processing. Ultrasound is a fast, non-invasive, inexpensive, and widely available imaging modality that has great potential for studying body composition. Ultrasound can directly measure muscles, organs, viscera, and subcutaneous fat in the abdomen and various parts of the body, overcoming some of the limitations of anthropometric assessment and other imaging techniques. Ultrasonography may play a role in characterizing body composition and examines four key topics, namely abdominal fat compartment, SAT, skeletal muscle, and liver. Typically used to evaluate: Pregnancy, Conditions of the heart and blood vessels, Pelvis and abdomen, Pain, swelling, and/or infection, Help guide certain procedures such as injections or biopsies [4].

1.4. What is Medical Image Segmentation:

Image segmentation is the process of separating a picture into various parts with the goal of preserving image quality throughout the classification. This method is also used to identify the boundaries of components of a picture. This method gives labels for different parts of the image according to the intensity and qualities of the pixels, capturing unique properties such as amplitude and similarity. Image segmentation is widely used in a range of professions, including clinical applications that produce a 3D contour of the human body. It also helps with machine perception, evaluating dangerous diseases, measuring tissue volume, performing structural and functional studies, applying 3D rendering techniques, visualizing virtual reality, analyzing anomalies, and identifying and recognizing objects.



Large bowel – *Small bowel* – *Stomach*

Figure 3. Medical Image Segmentation.

The dataset consists of 3 classes: the stomach, small bowel, and large bowel.

1.5. Methods Used in Medical Image Segmentation:

In this section, we'll briefly cover 4 types of segmentation modes you can use for medical imaging:

1.5.1. Threshold-based Segmentation:

Thresholding is a simple image segmentation method that divides pixels into classes based on histogram intensity aligned to a fixed value or threshold. In low-noise images, threshold values can stay constant, but in noisy images, a dynamic approach is more effective. Greyscale images are typically divided into two segments based on their relationship to the threshold value. Two common approaches are:

- **Global thresholding:** for image segmentation divides images into foreground and background regions, with a threshold value to separate the two.

- **Adaptive Thresholding:** Demarcates foreground and background using locally applied thresholds that depend on image characteristics.

1.5.2. Region-based Segmentation:

Segmentation by regions starts with one or more “starting pixels”. The region creation algorithms group together neighboring pixels with similar characteristics. Algorithms can be agglomerative or divisive.

1.5.3. Edge-based Segmentation:

Edge-based segmentation discovers and separates an image's edges from the background. AI technologies can identify changes in light or color values and use them to designate item boundaries in photos.

One approach is the Canny edge detection strategy, which uses a Gaussian filter, nonmaximum suppression to reduce the edges, and loop thresholding to eliminate weak edges [5].

Another approach, known as Sobel, includes computing an image's gradient magnitude and direction with a Sobel operator, which is a convolution kernel that separates horizontal and vertical edge information.

1.5.4. Machine Learning-based Segmentation:

With the advent of deep learning, convolutional neural networks (CNNs) have revolutionized medical image segmentation. Models like U-Net, SegNet, and Mask R-CNN have shown remarkable performance in segmenting medical images. These models learn from large datasets and can handle complex structures, achieving high accuracy and robustness.

1.6. The Benefits of Medical Image Segmentation:

One of the primary advantages of medical picture segmentation is that it enables more exact examination of anatomical data by separating only the relevant areas. Some operations, such as implant design, necessitate the segmentation of specific structures, such as those in the hip or knee. Furthermore, segmentation offers the added benefit of eliminating unwanted details from scans, such as air, while enabling the isolation of different tissues like bone and soft tissues. By utilizing various software processing options, researchers and clinicians can generate a series of segmented masks that are readily available for further analysis [6].

1.7. Challenges and Future Directions:

The accuracy of segmentation algorithms in medical image segmentation is impacted by various challenges. These challenges include the variability in image acquisition, the presence of noise and artifacts, as well as inter-patient anatomical variations.

To tackle these obstacles, it is imperative to conduct additional research in order to create strong algorithms, integrate multi-modal imaging, and delve into advanced methods such as 3D segmentation. Moreover, the fusion of artificial intelligence and machine learning methodologies shows great potential in enhancing both the accuracy and efficiency of segmentation.

1.8. Using U-Net:

When considering image segmentation in machine learning, the model that we think to utilize is U-Net. This is a significant improvement in performance compared to traditional techniques. U- Net is a convolutional neural network that has a wide range of applications, including medical imaging, autonomous driving, and satellite imaging. However, it is important to understand how this model performs segmentation because all new architectures after U-Net were developed based on the same intuition. We will take a deeper look at how U-Net performs image segmentation.

Chapter 02 : U-net image segmentation

2.1. Introduction :

Thanks to recent advances in deep learning in computer vision within the past decade, deep learning has been increasingly utilized in the analysis of medical images. While the use of deep learning in computer vision has seen rapid growth in many different fields, it still faces some challenges in the medical imaging field. One such technique that will be discussed in this chapter will be the U-net, a deep learning technique widely adopted within the medical imaging community.

2.2. U-Net Architecture:

U-net is a convolutional neural network designed for image segmentation, especially for biomedical image such as cell segmentation, organ localization and medical image analysis. It is one of the original deep learning segmentation models, was first developed and implemented in 2015. Convolutional neural networks are usually employed for image classification tasks, where the output to an image is a single class label, but when it comes to biological image processing tasks, it is not just necessary to determine whether there is a medical condition, but also to localize the area of infection; it means each pixel is expected to have its own class labeling. The U-Net is also utilized in other GAN variations, including the Pix2Pix generator.

U-net's architecture name inspired by its U-shaped design. The model architecture is quite simple, contains two main paths: encoder path (for downsampling) and decoder (for upsampling) connected by the connections path [7].

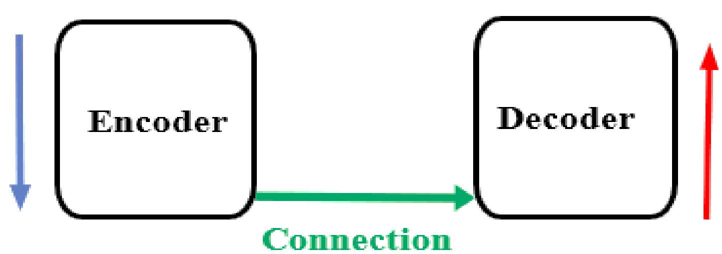


Figure 3.1. U-net architecture.

2-1-Encoder:

The encoder represents the first half of the architectural diagram. It is generally a pre-trained classification network, such as VGG/ResNet, that uses convolution blocks followed by maxpool downsampling to encode the input picture into feature representations at various levels.

- Made up of convolutional 3 × 3 Relu to extract features and minimize spatial dimensions.

- Max pooling 2×2 to downsample the feature maps and reduce computational demands.

2-2- Decoder:

The decoder represents the second part of the architecture. The objective is to semantically project the encoder's learned discriminative features (lower resolution) onto the pixel space (higher resolution) to achieve dense classification. The decoder includes of upsampling and concatenation, followed by standard convolution operations.

- Access high resolution information from earlier stage.

- Upsampling layers by convolution 3 × 3 Relu.

That can create a segmentation map that has the same dimensions as the input image by recovering the spatial informations losted during the encoding process.

2-3-Skip connections:

Skip connections bridge the distance between the encoder and decoder by:

- Facilitate the transmission of fine features during the upsampling process.

- Enable the decoder to incorporate both local and global context information to improve segmentation masks.

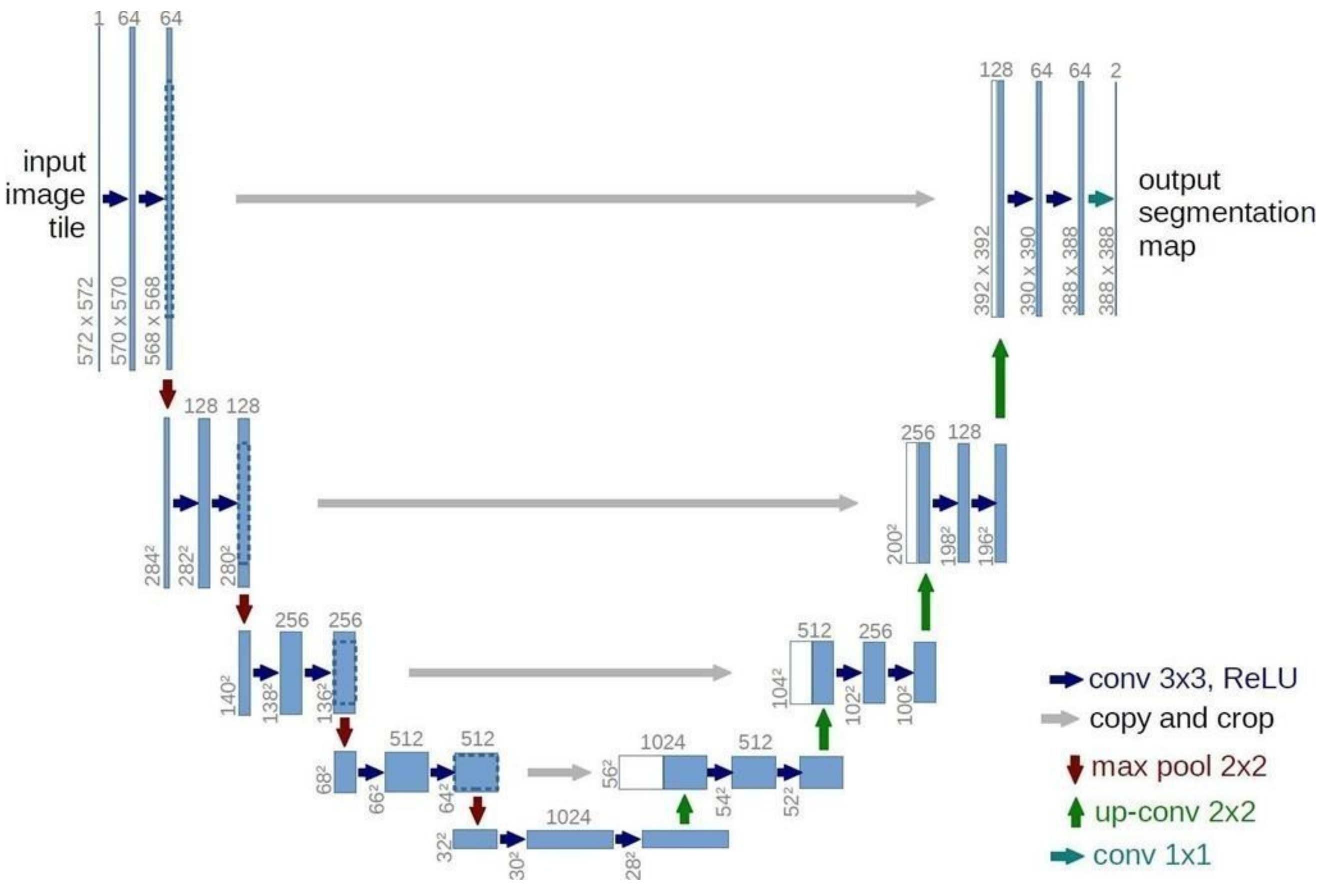


Figure 3.1. U-net architecture

2.3. Medical Image Segmentation Based On U-net:

Nowadays, the new technique called medical image segmentation gives surgeons a virtuel vision. It offers comprehensive analysis and easy tests, making it an extremely useful tool in the medical field.

CNNs became known as the foundation of deep learning for medical imagine segmentation. Their capacity to deal with visual data via hierarchical feature extraction and representation learning makes them very effective in detecting complex patterns in medical photos.

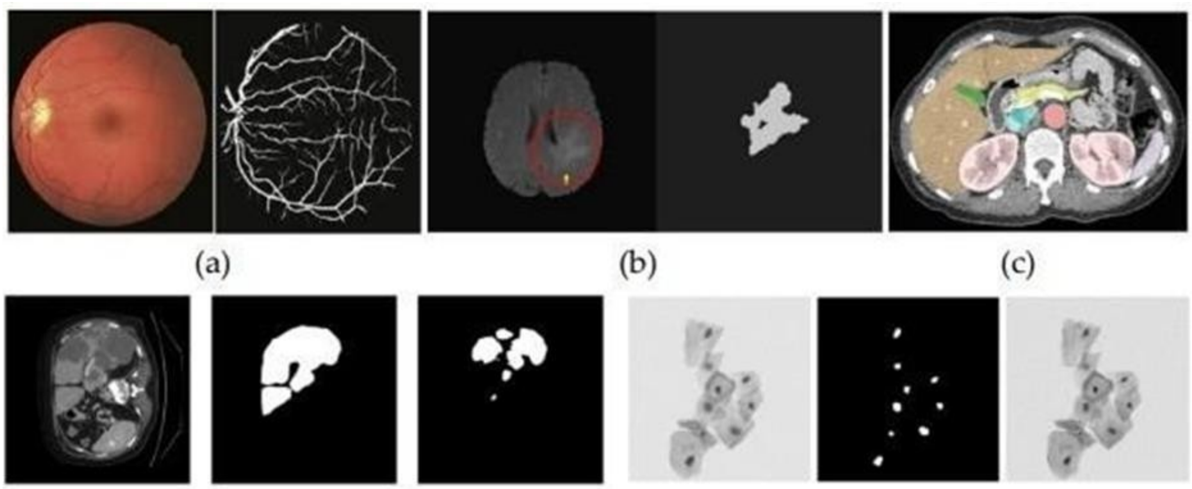


Figure 4. Examples of U-net applications.

2.3.1 U-net segmentation for tumor segmentation:

Using a U-net architecture for heart tumor image segmentation is popular and useful method in medical image segmentation. So, this are the steps of this method:

3-1-1- Data collection and preprocessing:

- Collect a dataset of CT heart with ground truth masks that localized heart.

- Preprocess image by resizing them to a uniform resolution and adding more information to make it more variable

- we use this dataset from kaggle (CT Heart Dataset (kaggle.com))

3-1-2-Model architecture:

- Implement the U-net model (encoder downsamples, decoder upsamples).
-Insert skip connections.

3-1-3-Training:

- Divide the dataset into sets for testing, validation, and training.

- Train the U-net model on the training set applying a loss function such dice coefficient loss or binary cross-entropy.

- Notice the model's performance on the validation set and modify the hyper- parameters to avoid overfitting.

3-1-4-Evaluation:

- Use metrics like the dice coeficient, and pixel wise accuracy to evaluate the trained model on the test set.

- Visualize the segmentation accuracy qualitatively by comparing the model's predictions with the ground truth masks.

2.4. Chalenges:

Implementing AI in healthcare poses challenges, including data privacy, biases in training data, and the interpretability of AI-driven decisions. Ensuring transparent, ethical, and responsible AI deployment is critical to maintaining patient trust and safety.

Implementing AI in healthcare presents problems such as data privacy, biases in training data, and the interpretability of AI-driven judgments. Ensuring transparent, ethical, and responsible AI deployment is important to maintain patient confidence and safety.

2.5. Conclusion:

Medical image segmentation, connected with deep learning algorithms, has been recognized as an interactive changer in healthcare. From radiology to cardiology, the effect on diagnosis, treatment planning, and patient outcomes is apparent.

In this chapter We studied the U-net definition, architecture and method and we hope to provide a starting point for researchers who wish to explore U-net. The investigation concludes that U-net based architecture is extremely innovative and useful for medical image processing

Chapter 03: Modified U-net (Law’s-Net).

3.1. Introduction:

The gray level distribution of an image is assigned to the set of pixels; typically, this information is unable to serve as a strong segmentation requirement for textured images. This is accomplished by extracting pertinent features with a high discriminating power from the image that describe each texture with regard to a single pixel or a group of pixels in a specific region.

In this chapter, we employed one of most effective space techniques K. I. Laws's (1980), which was developed specifically for segmenting textured images. The idea behind this technique is to measure the energy of a textured image at both the microscopic and macroscopic levels in a field of space in order to take advantage of the micro periodicities that exist within it. We designed our modified U-Net architecture.

3.2. Law’s masks:

Laws has created effective filters (masks) that, when used on an image, can extract specific kinds of almost periodicities that are typically present in textured images that are susceptible to visual structures like borders, waves, stains, etc.

They are produced by combining two of the following four vectors:

L5 = [1 4 6 4 1]; Level detection.

E5 = [-1 -2 0 2 1]; Edge detection.

S5 = [-1 0 2 0 -1]; Spot detection.

R5 = [1 -4 6 -4 1]; Ripple detection.

W5= [-1, 2, 0, -2, 1]; wave detection.

$$E5L5 = \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad R5R5 = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$E5S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -2 & 0 & 4 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 2 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix} \quad L5S5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

Figure 6. Four of Laws most successful masks.

The mentioned masks are convolved with the original image to create a series of images which are themselves passed to a second level of processing, referred to “macrostatistic" [Laws79]. This consists of a moving window estimation of the energy within the images. Thus, Laws’ feature measures estimate the energy within the passband of their associated filters and he therefore called his operators "texture energy measures" [8].

3.3. Laws Filters Application:

We have applied Laws filters to a CT image [12]. Laws filters are a set of spatial filters used for texture analysis, as mentioned. These filters capture variations in pixel intensities across different directions and scales.

We are going to explain each filter and his effects on image we start Filtered_image_E5L5:

The first image (Filtered_image_E5L5) result shows that it has been filtered using both the E5 and L5 filters. The resulting image combines the effects of both filters. We will provide an explanation for both of them: The E5 filter enhances the edges and boundaries within the image, making them more distinct and prominent. Meantime, the L5 filter reduces texture irregularities and accentuates regions of uniformity, thereby reducing noise and accentuating areas with consistent levels of intensity.

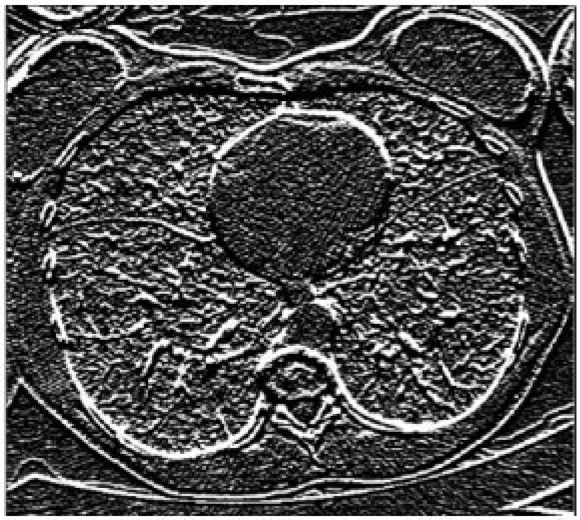


Figure 7.1. Filtered_image_E5L5.

The second image (Filtered_image_E5S5) result shows that it has been filtered using both the E5 and S5 filters. The resulting image combines the effects of both filters. The E5 filter enhances the edges and boundaries, making them sharper and more noticeable. Meanwhile, the S5 filter targets smaller areas and delicate textures within the image, effectively capturing complex details and patterns that may have been less visible in the original image.

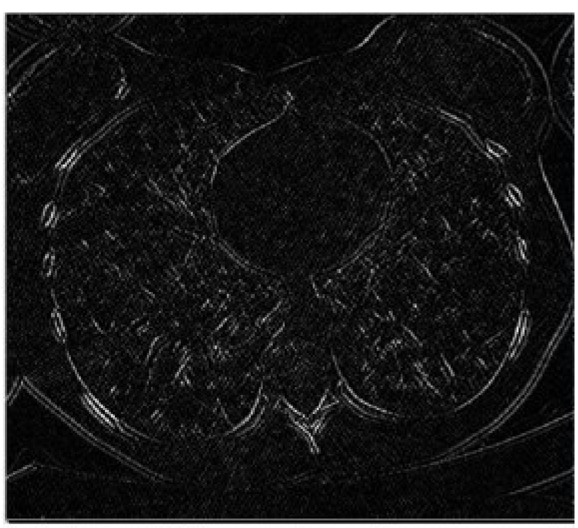


Figure 7.2. Filtered_image_ E5S5

The third image (Filtered_image_L5S5) result shows that it has been filtered using both the "L5" and "S5" filters. The resulting image combines the effects of both filters. The L5 filter utilized for edge detection and highlights areas of rapid intensity change in the image, which typically correspond to edges. It can enhance the edges in an image, making them more pronounced. Meanwhile, the S5 filter targets smaller areas and delicate textures within the image, effectively capturing complex details and patterns that may have been less visible in the original image.

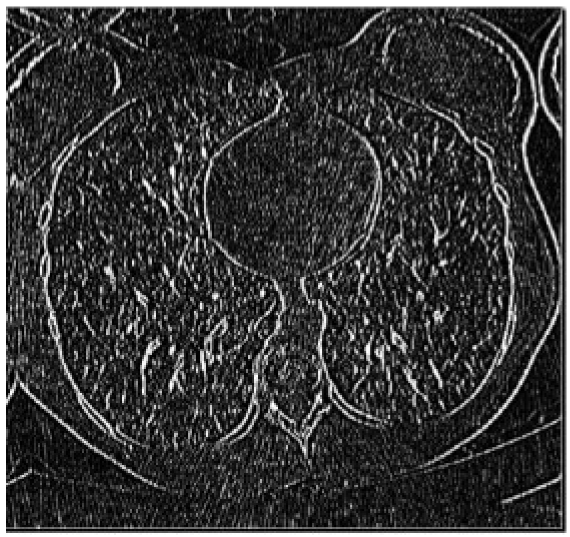


Figure 7.3. Filtered_image_ L5S5

The fourth image (Filtered_image_R5R5) result shows that it has undergone a double filtration process utilizing the "R5" filter. The outcome of this process, As the filter is applied repeatedly, pixel values are averaged with those of neighboring pixels within a 5x5 kernel, resulting in a gradual decrease in high-frequency details and noise. Consequently, the intensity variations within the image are smoothed out, potentially leading to a reduction in the sharpness of edges when compared to the original image. This, in turn, gives rise to a softer appearance of edges. Furthermore, with each iteration of the filtering process, noise within the image is progressively diminished as neighboring pixel values are averaged together.

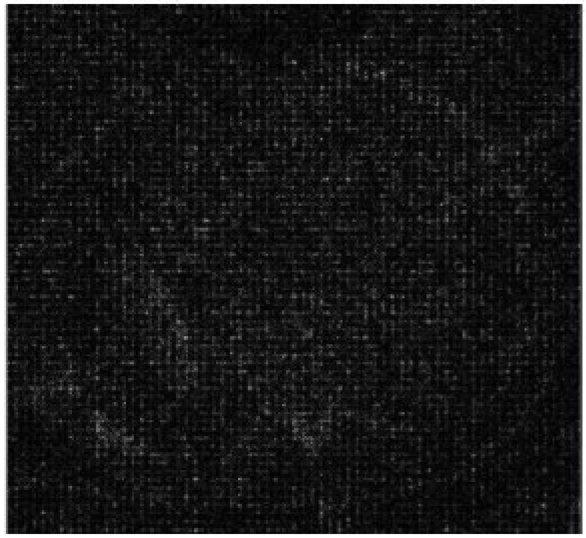


Figure 7.4. Filtered_image_R5R5.

3.3. Modified U-Net architecture:

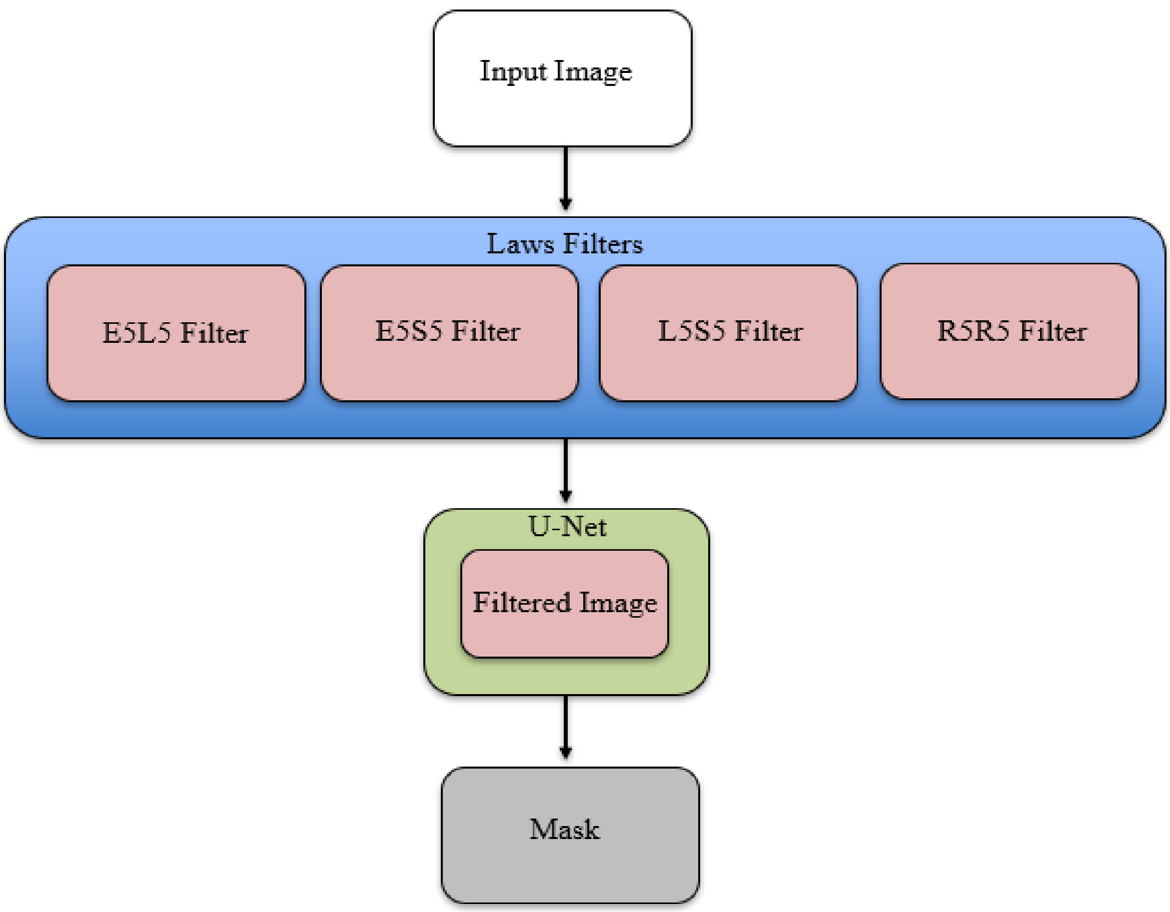


Figure 8. Modified U-Net Architecture (Mask production).

In this work, we propose a modified architecture based on the popular U-Net convolutional neural network for image segmentation tasks. Our modification aims to address specific limitations within the input image of this architecture, improve upon certain aspects to produce a mask more effective than the original U-Net architecture mask.

3.3. Conclusion:

In this chapter, we studied Law's Masks and Filters. We applied the Laws filters to a medical image to obtain a filtered image. We then observed the effects on the input image and explained the function of each filter. In essence, our U-Net modified architecture based on filtered images provides sharper segmentation, noise reduction, versatile feature representation, and efficient adaptation, making it a robust solution for various image segmentation challenges.

Chapter 04: Implementation and results

4.1. Introduction:

In this chapter we focus on the implementation of our deep learning model using Laws-Net architecture and evaluating its application on a medical image dataset.

4.2.1. Programming Language Overview "Python":

Python is a high-level language. A single line of code may be used to perform complicated processes in some cases, by omitting details such as data management, representation in computer memory, basic arithmetic, or binary operations. A high-level language makes program development easier, increasing productivity (often at the expense of execution speed). It is open source and therefore free for everyone, including businesses. This is a popular programming language for which we can easily find documentation and numerous examples of its use in various applications on the internet. Furthermore, the community creates and maintains a large number of Python-compatible libraries, which enhance productivity when writing a program, especially in the fields of mathematics and data sciences.

4.2.2. Programming Language Overview "GOOGLE COLAB":

We have used Google Colab, or Colaboratory, which is a free cloud service offered by Google. It is based on Jupyter Notebook and is designed for machine learning training and research. This platform helps you to train machine learning models directly in the cloud, as well as develop and run Python code in a collaborative environment, all without installing anything on your PC other than a browser. It is useful for machine learning, data analysis, and research projects.

- **Jupyter Notebook:** is an essential tool for data scientists. It is an open-source web tool that allows users to create and share documents containing code (which can be executed directly in the document), equations, graphics, and text. This program enables you to perform data processing, statistical modeling, data visualization, machine learning, and more.

- **The cloud version of Jupyter Notebook:** To use Google Colab, go to Google Drive, then click on "new," then "more," and finally "Google Colaboratory"[9].

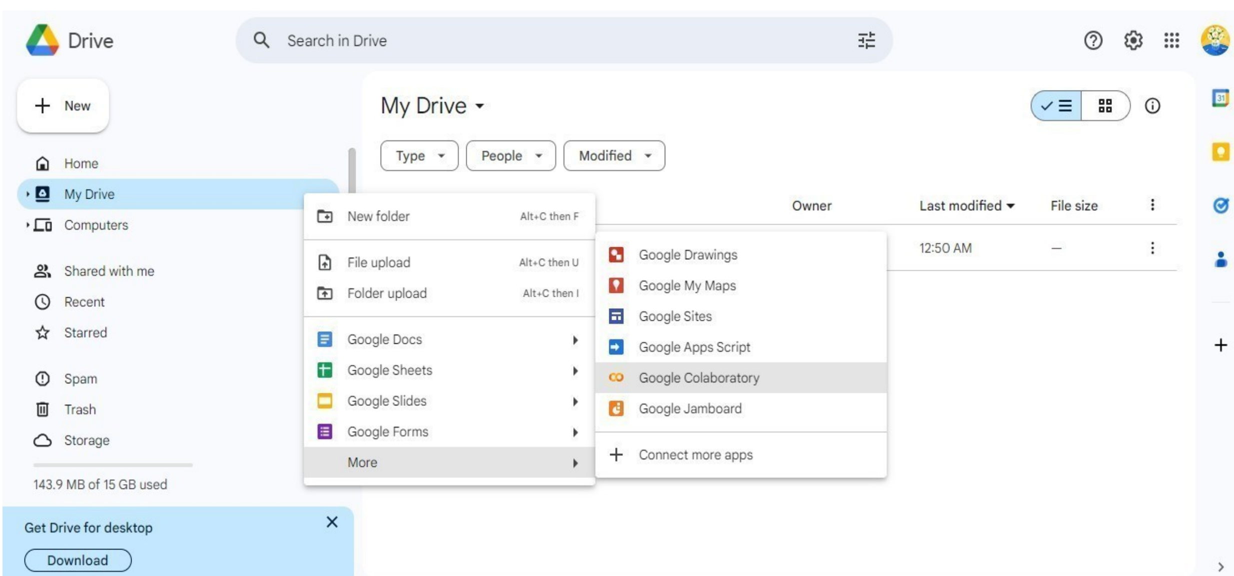


Figure 9.1. Google Colab.

You may also go directly to the Google Colab interface by clicking here. The window will appear as seen below:

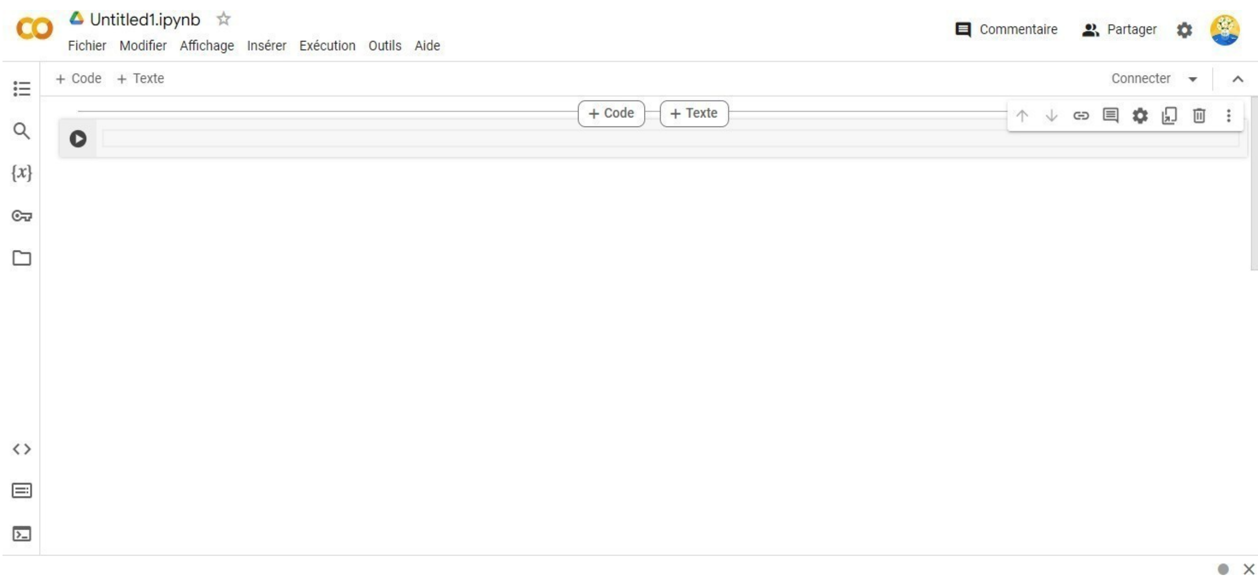


Figure 9.2. Google Colab.

4.3. Test approach:

4.3.1. Method:

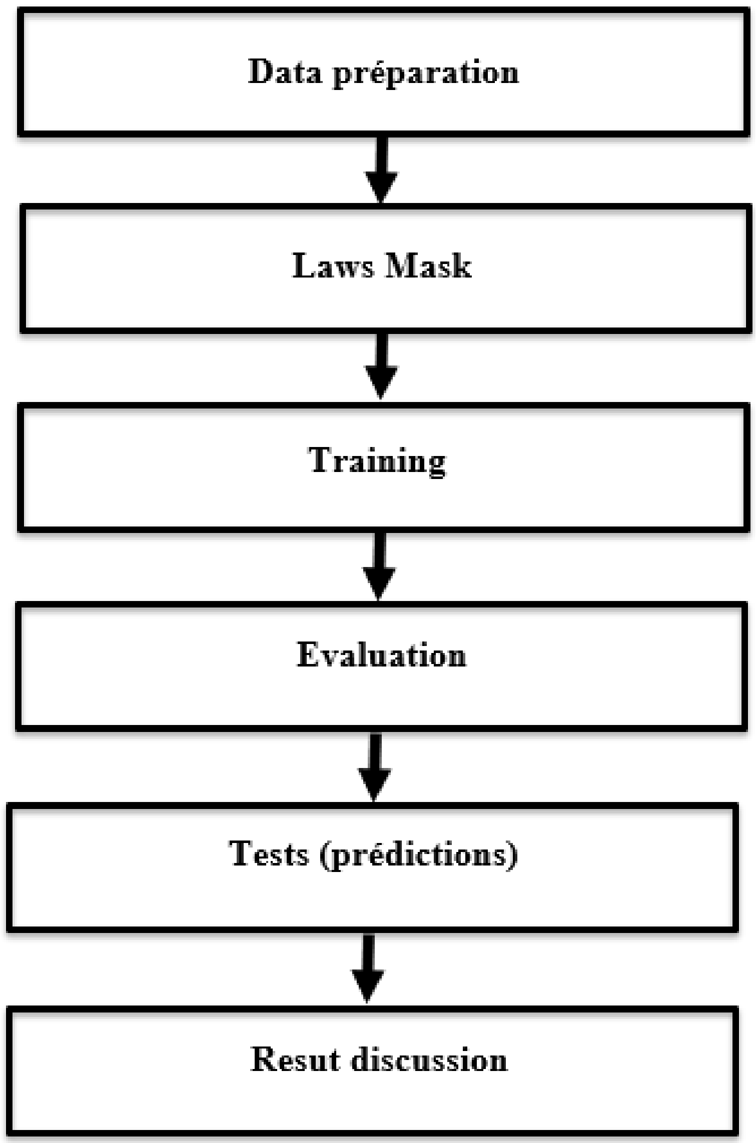


Figure 10. Implementation process

4.3.2. Data preparation:

In this study we use a biomedical image of CT heart to segment the assessment of cardiac anatomy and specifically the coronary arteries . with the Laws-Net segmentation, and this is the link of this data: CT Heart Dataset (kaggle.com) The dataset contains 8104 images with size (512 x 512 x 4) in three folders: train, test and valid. The train folder contains both the images and masks, with 2532 images and 2532 masks. Also, the test folder contains 832 images and masks.

4.3.2.1. Data augmentation:

Data augmentation is a technique used to increase the diversity of your training set by applying random transformations such as flipping and rotation.

Example: Original image and its corresponding mask.

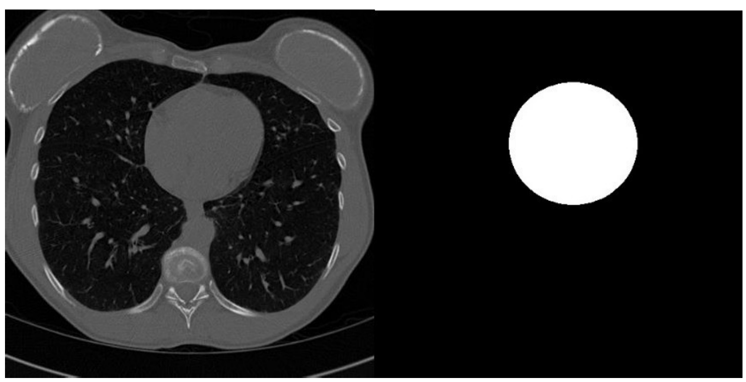


Figure 11.1. Original image and its corresponding mask.

2.1.1. The vertical flip: Vertical flip basically flips both rows and columns vertically.

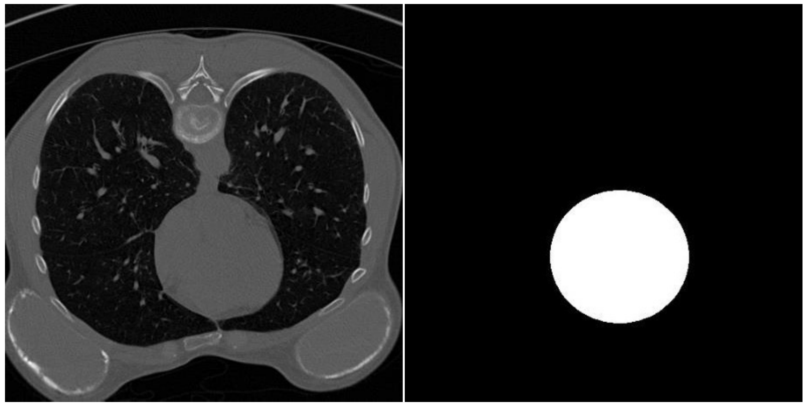


Figure 11.2. vertical flip image.

2.1.2. The Horizontal flip: Horizontal flip basically flips both rows and columns horizontally.

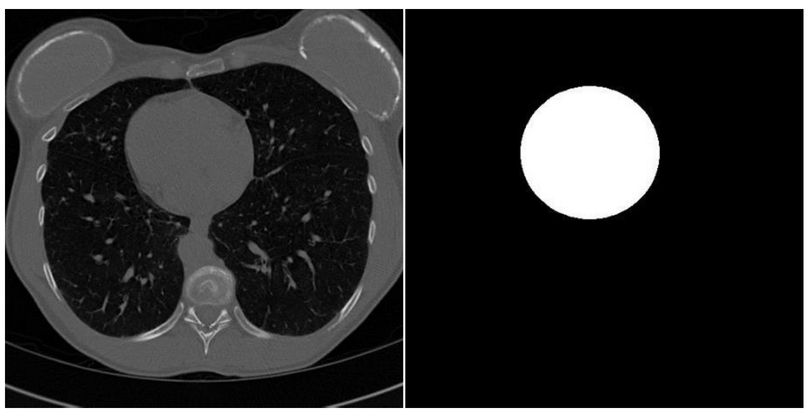


Figure 11.3. Horizontal flip image.

2.1.3. Rotation (45°): is a technique used to increase the size of data by rotating images by a certain angle:

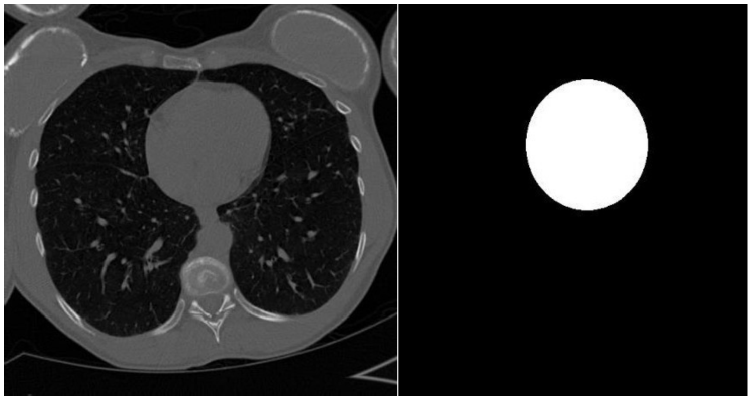


Figure 11.4. Rotation image.

4.3.3. Creation of Laws masks:

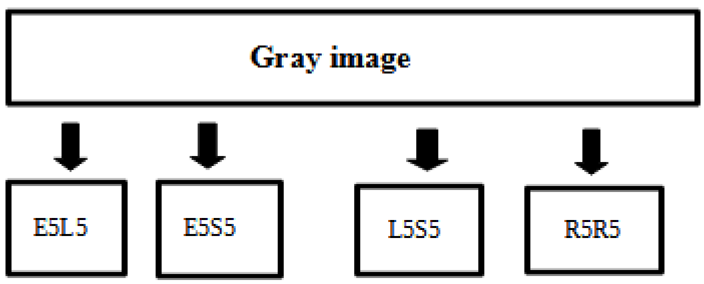


Figure 12. Creation of Laws masks.

3.3.1. Application of Filters on Images:

```
def createLawsMasks():
  # Define the Laws filter kernels
  L5 = np.array([1, 4, 6, 4, 1])
  E5 = np.array([-1, -2, 0, 2, 1])
  S5 = np.array([-1, 0, 2, 0, -1])
  W5 = np.array([-1, 2, 0, -2, 1])
  R5 = np.array([1, -4, 6, -4, 1])

  laws_filters = [];
```

```
  laws_filters.append(np.outer(E5, L5))
  laws_filters.append(np.outer(R5, R5))
  laws_filters.append(np.outer(E5, S5))
  laws_filters.append(np.outer(L5, S5))

  return laws_filters

def createLawsPlanesFromImage(gray_image, laws_filters):
  # Convolve the image with each of the Laws filters
  convolved_images = []
  for filt in laws_filters:
      convolved_images.append(cv2.filter2D(gray_image, -1, filt))

return convolved_images
```

The four laws masks will be applied to each image with its augmented versions. The effect of application of the Laws filters is illustrated in the following figure:

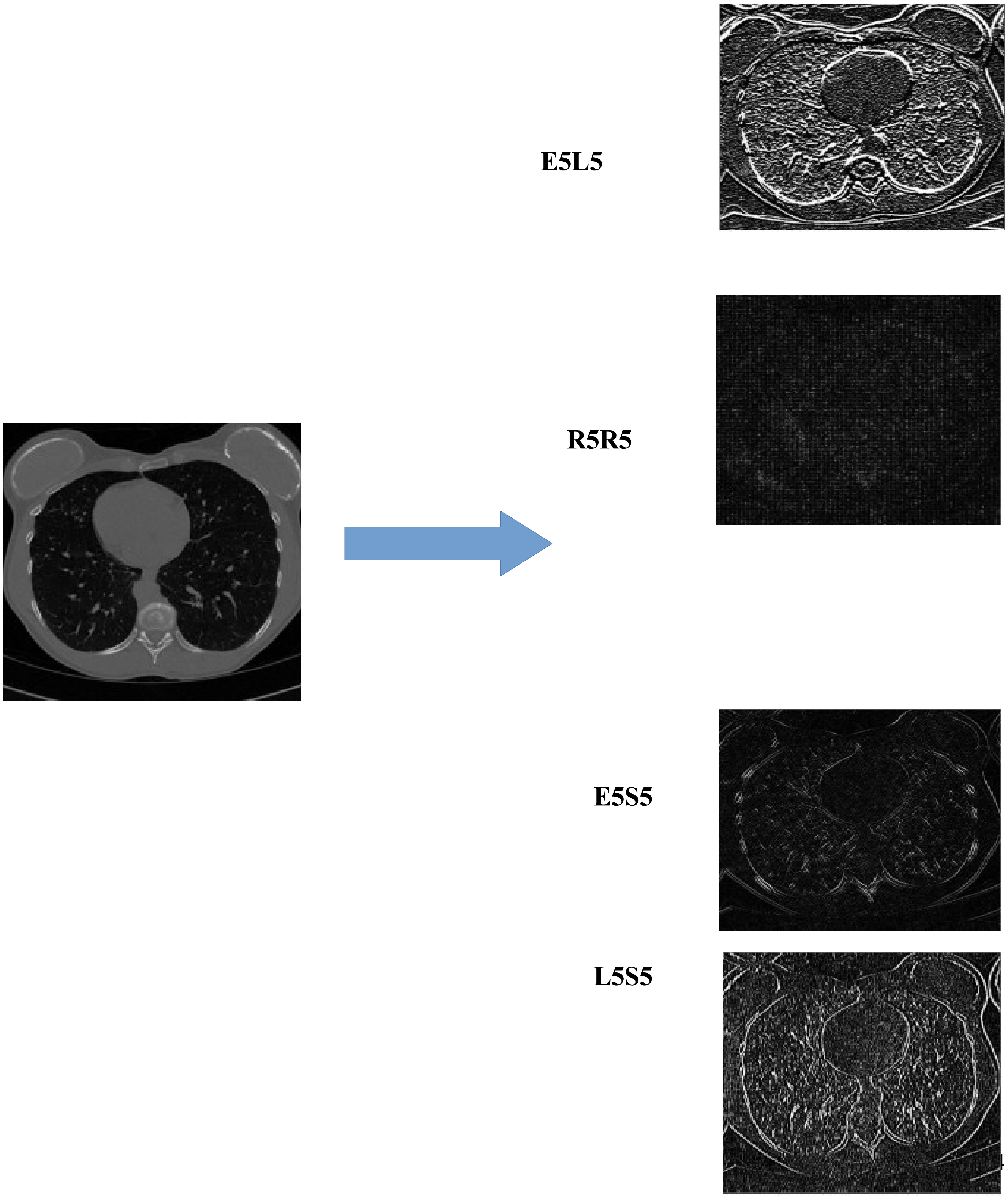

E5L5
R5R5
E5S5
L5S5

Figure 13. Filtered Images.

4.3.4. Training:

Use a GPU-enabled environment to train the model with backpropagation, using an appropriate loss function like cross-entropy for pixel-wise classification. in this part we do the training of the model with 8104 images and masks using just 2 of epoch and batch size =8 because it takes long time with limited GPU(T4). Our train model takes 3142s 765ms/step in the first epoch, and 6116s 2s/step(1h6min8s) in the second epoch.

This is the link of our source code : https://colab.research.google.com/drive/10xW-MxHgQe1lFMUDM9zqgauotpt6WbeL?usp=sharing

3.4.1Training Code Implementation:

```
if __name__ == "__main__":
    """ Seeding """
    np.random.seed(42)
    tf.random.set_seed(42)

    """ Directory for storing files """
    create_dir(work_dir +"files")

    """ Hyperparameters """
    batch_size = 2
    lr = 1e-4
    num_epochs = 2
    model_path = os.path.join(work_dir +"files", "model.h5")
    csv_path = os.path.join(work_dir +"files", "data.csv")

    """ Dataset """
    dataset_path = os.path.join(tiff_dataDir)
    train_path = os.path.join(dataset_path, "train")
    valid_path = os.path.join(dataset_path, "valid")

    train_x, train_y = load_data(train_path)
    train_x, train_y = shuffling(train_x, train_y)
    valid_x, valid_y = load_data(valid_path)

    print(f"Train: {len(train_x)} - {len(train_y)}")
    print(f"Valid: {len(valid_x)} - {len(valid_y)}")
```

```python
print(f"Train: {len(train_x)} - {len(train_y)}")
print(f"Valid: {len(valid_x)} - {len(valid_y)}")

train_dataset = tf_dataset(train_x, train_y, batch=batch_size)
valid_dataset = tf_dataset(valid_x, valid_y, batch=batch_size)

""" Model """
model = build_unet(input_shape) #(H, W, 3)
metrics = [dice_coef, iou, Recall(), Precision()]
model.compile(loss=dice_loss, optimizer=Adam(lr), metrics=metrics)

callbacks = [
    ModelCheckpoint(model_path, verbose=1, save_best_only=True),
    ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=10, min_lr=1e-7, verbose=1),
    CSVLogger(csv_path),
    TensorBoard(),
    EarlyStopping(monitor='val_loss', patience=50, restore_best_weights=False),
]

model.fit(
    train_dataset,
    epochs=num_epochs,
    validation_data=valid_dataset,
    callbacks=callbacks,
    shuffle=False
)
```

4.3.5Training Creteria:

This is the result of training:

epoch	dice_coef	iou	loss	lr	precision	recall	val_dice_coef	val_iou	val_loss	val_precision	val_recall
0	0.43291	0.360524	0.56709	1.00E-04	0.383219	0.815724	0.314278	0.259796	0.685722	0.338898	0.980397
1	0.518558	0.463106	0.481442	1.00E-04	0.521577	0.864224	0.364973	0.327996	0.635027	0.793405	0.896627

Table 2. Training performance criteria for two epochs

4.3.4. Evaluation:

The segmentation process is a critical task in image processing and computer vision tasks, aiming to partition an image into meaningful segments, often to identify objects or boundaries within scenes. Evaluating the performance of these segmentation models is crucial to ensure their reliability and effectiveness in real-world applications. In the following section we give the most used metrics to evaluate deep image segmentation models [10].

3.5.1 Evaluation Code Implementation:

```
if __name__ == "__main__":
    """ Seeding """
    np.random.seed(42)
    tf.random.set_seed(42)

    """ Directory for storing files """
    create_dir(tiff_dataDir + "results")

    """ Loading model """
    with CustomObjectScope({'iou': iou, 'dice_coef': dice_coef, 'dice_loss': dice_loss}):
        model = tf.keras.models.load_model(model_path +"model.h5")

    """ Load the dataset """
    test_x = sorted(glob(os.path.join(tiff_dataDir, "valid", "image", "*")))
    test_y = sorted(glob(os.path.join(tiff_dataDir, "valid", "mask", "*")))
    print(f"Test: {len(test_x)} - {len(test_y)}")

    """ Evaluation and Prediction """
    SCORE = []
    for x, y in tqdm(zip(test_x, test_y), total=len(test_x)):
        """ Extract the name """
        name = x.split("/")[-1].split(".")[0]

        """ Reading the image """
```

```
        ret, images = cv2.imreadmulti(x, [], cv2.IMREAD_ANYCOLOR)

        frames = np.array(images)
        frames = frames.astype(np.float32)/255.0
        x = np.moveaxis(frames, 0, 2)
        x = np.expand_dims(x, axis=0)

        """ Reading the mask """
        mask = cv2.imread(y, cv2.IMREAD_GRAYSCALE)
        y = mask/255.0
        y = y > 0.5
        y = y.astype(np.int32)

        """ Prediction """
        y_pred = model.predict(x)[0]
        y_pred = np.squeeze(y_pred, axis=-1)
        y_pred = y_pred > 0.5
        y_pred = y_pred.astype(np.int32)

        """ Saving the prediction """
        save_image_path = tiff_dataDir +  f"results/{name}.png"
        save_results(image, mask, y_pred, save_image_path)

        """ Flatten the array """
        y = y.flatten()
        y_pred = y_pred.flatten()
```

```
        """ Calculating the metrics values """
        acc_value = accuracy_score(y, y_pred)
        f1_value = f1_score(y, y_pred, labels=[0, 1], average="binary", zero_division=1)
        jac_value = jaccard_score(y, y_pred, labels=[0, 1], average="binary", zero_division=1)
        recall_value = recall_score(y, y_pred, labels=[0, 1], average="binary", zero_division=1)
        precision_value = precision_score(y, y_pred, labels=[0, 1], average="binary", zero_division=1)
        SCORE.append([name, acc_value, f1_value, jac_value, recall_value, precision_value])

    """ Metrics values """
    score = [s[1:]for s in SCORE]
    score = np.mean(score, axis=0)
    print(f"Accuracy: {score[0]:0.5f}")
    print(f"F1: {score[1]:0.5f}")
    print(f"Jaccard: {score[2]:0.5f}")
    print(f"Recall: {score[3]:0.5f}")
    print(f"Precision: {score[4]:0.5f}")

    df = pd.DataFrame(SCORE, columns=["Image", "Accuracy", "F1", "Jaccard", "Recall", "Precision"])
    df.to_csv(model_path + "score.csv")
```

3.5.2 Evaluation Metrics:

There are four rates upon which most evaluation metrics are built:

- True positive (TP): the number of observations that are correctly classified. For instance, if the classification is about a pixel is belonging to a tumoral class or not. If the pixel is tumoral and classified as tumoral, it is TP.

- False positive (FP): the number of observations that are not correctly classified. For instance, if the object is not tumoral and classified as tumoral, it is FP.

- True negative (TN): the number of observations that are classified. For instance, if the object is not tumoral and classified as not tumoral, it is TN.

- False negative (FN): the number of observations that are not classified.

For instance, if the object is nottumoral and is classified as tumoral, it is FN.

$$\text{Accuracy} = (TP + FP)/(TP + FP + TN + FN)$$

$$\text{Sensitivity} = (TP)/(TP + FN)$$

3.5.1.1. Pixel Accuracy:

Pixel accuracy measures the ratio of correctly classified pixels to the total number of pixels. It is one of the simplest metrics for evaluating segmentation models.

Pixel Accuracy = Number of Correct Pixels / Total Number of Pixels

While easy to compute, pixel accuracy can be misleading, especially in datasets with class imbalance, where larger classes dominate the accuracy score.

3.5.1.1. Intersection over Union (IoU):

IoU, also known as the Jaccard Index, calculates the overlap between the predicted segmentation and the ground truth.

IoU=Intersection of Prediction and Ground Truth / Union of Prediction and Ground Truth

IoU is a more robust metric than pixel accuracy, providing a clearer picture of segmentation quality. However, it can still be affected by class imbalance.

3. 5. 1. 1. Dice Coefficient:

The Dice Coefficient, similar to IoU, measures the similarity between the predicted segmentation and the ground truth.
Dice_Coefficient = 2× | Prediction ∩ Ground Truth| / |Prediction |+| Ground Truth | This metric is particularly useful in medical imaging, where accurate boundary detection is critical.

3.5.2.4 Precision, Recall and F1 Score:

• Precision measures the ratio of correctly predicted positive observations to the total predicted positives.
Precision= True Positives/ (True Positives + False Positives)
• Recall measures the ratio of correctly predicted positive observations to all observations in the actual class.
Recall=True Positives / (True Positives+False Negatives) • F1 Score is the harmonic mean of precision and recall. F1_Score=2×Precision×Recall/(Precision+Recall).

These metrics provide a balanced evaluation, especially in scenarios with imbalanced classes.

This are some results of evaluation:

	A	B	C	D	E	F	G
1		Image	Accuracy	F1	Jaccard	Recall	Precision
2	0	valid_100051_1-026	1	1	1	1	1
3	1	valid_100051_1-030	1	1	1	1	1
4	2	valid_100051_1-044	0.9959182739	0	0	1	0
5	3	valid_100051_1-045	0.9977798462	0	0	1	0
6	4	valid_100051_1-049	0.9802474976	0	0	1	0
7	5	valid_100051_1-050	0.9735412598	0	0	1	0
8	6	valid_100051_1-052	0.9734916687	0	0	1	0
9	7	valid_100051_1-057	0.9903755188	0.8996300274	0.8175704989	0.8455096089	0.9611526692
10	8	valid_100051_1-068	0.9887313843	0.9054296325	0.8272009359	0.8715562404	0.9420425022
11	9	valid_100051_1-070	0.9805641174	0.809240331	0.6796000503	0.679984899	0.9991678994

Figure 14. Some Results of Evaluation.

4.3.5. Test predictions:

The following figures represent some output maps (predicted masks) after segmentation applied on test data:

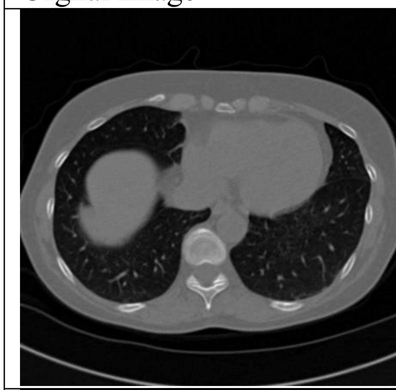
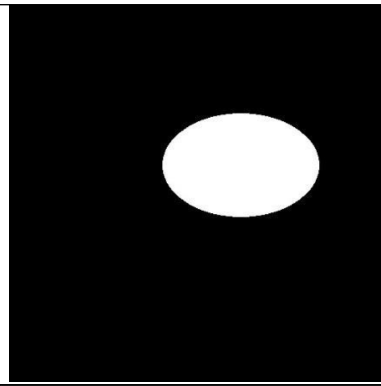
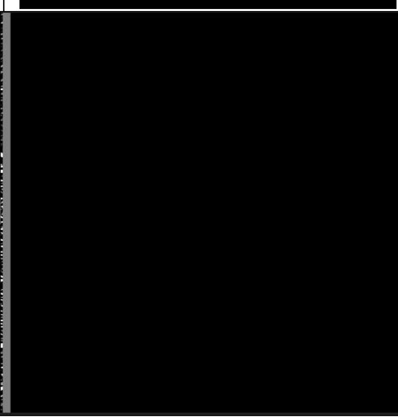
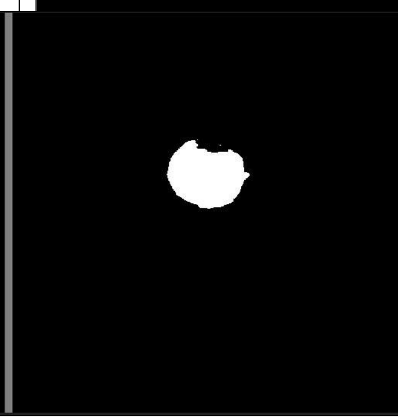
Orgnal Image	Ground truth Mask	Predicted Mask
		
		

Figure 15. Some Output maps after segmentation applied on test data.

4.3.6. Result discussion:

The segmentation performances of the different compilation metrics of Laws-Net model are presented in table 3:

Metrics	Accuracy	Recall	Precision	F1	IoU
Values	0.98923	0.95514	0.72639	0.72145	0.69281

Table 3. The segmentation performances.

1- Accuracy: the value of accuracy (0.98923) indicates that the model predicts correctly a high percentage of all occurrences.
2- Recall: The value of Recall (0.95514) indicates that the model can identify 95.14% of the relevant instances (True positives).
3- Precision: the value of precision (0.72639) indicates that the model predicts a positive instance correct 72.63% of the time.
4- F1 score: the value of F1 (0.72145) indicates that the model is reasonably balanced performance between identifying positive instances and limiting false positives.
5- Jaccard index (iou): the value of iou (0.69281) indicates that the model measures the similarity between the predicted and actual segments, and a value close to 0.7 indicates a good overlap, though there is still room for improvement.

So, all these results indicate that our model have a good performance.

4.4. Conclusion:

In this chapter, we implemented state-of-the-art Laws-Net deep neural networks to automatically segment the assessment of cardiac anatomy and specifically the coronary arteries . This model receives a CT slice filtered as input and gives a binary mask as output that specifies the region of tumor. The results demonstrate that the Laws-Net model performs very well, achieving a dice coefficient of 0.5186 and an IoU of 0.4631.

We showed that our trained model achieved a high accuracy rate for detecting heart tumor. We report the model performance under various hyper-parameter settings, which can be helpful for future research to know the impact of different parameters on the final results.

General Conclusion:

Today, image processing is becoming increasingly important. Many segmentation algorithms and models have been developed to analyze images obtained by any process. It is therefore essential to evaluate these segmentation methods in order to measure their performance.

In this work, we used deep neural networks, specifically convolutional neural networks (CNNs), to automatically segment the tumor inside heart from CT images.

On the theoretical side, we introduced the deep learning technique in the field of medical image segmentation in general. We explained the role of each one of these components in the image segmentation process, then we talked in detail about one of the most popular and widely used deep learning architectures for segmenting medical images (U-Net architecture). Also, we used a modified U- Net (Law-Net) by applying the Laws filters on our data's images.

On the experimental side, we implemented this architecture using various image segmentation evaluation metrics to automatically detect and segment the areas of the tumor.

We noticed that the amount of data used to build the structure and some parameters played an important role in changing the performance of the decision model for automatic segmentation. Since the data used in this work is widely available, this made our built model achieve satisfactory results using various image segmentation evaluation metrics. A comparison between various image segmentation metrics of areas of tumor shows that all of these metrics perform very well.

The results obtained in this work represent promising prospects for the possibility of using deep learning to assist in an objective diagnosis of tumor through CT images of the heart.

Finally, our deep learning model can make the job of medical expert easier in detecting tumor.

References

[1] https://healthdirect.gov.au/x-rays.

[2] https://www.banglajol.info/index.php/BMRCB/article/view/57767.

[3] https://www.nibib.nih.gov/science-education/science-topics/magneticresonance-imaging-mri.

[4] Research in Medical Imaging Using Image Processing Techniques DOI: http://dx.doi.org/10.5772/intechopen.84360.

[5] Medical Image Segmentation: A Complete Guide | Encord.

[6] https://www.synopsys.com/glossary/what-is-medical-image-segmentation.html.

[7] Medical Image Segmentation [Part 1] — UNet: Convolutional Networks with Interactive Code | by Jae Duk Seo | Towards Data Science.

[8] http://www.macs.hw.ac.uk/texturelab/files/publications/phds_mscs/MJC/MJC_Thesis.pdf.

[9] https://github.com/googlecolab/colabtools.

[10] Precision and Recall in Classification Models. 2024, builtin.com/data-science/precision-and-recall accessed 4 Jun. 2024.

[11] https://www.jeremyjordan.me/evaluating-image-segmentation-models/.

[12]https://www.kaggle.com/datasets/f41e0bab640002775b00e050b81a1144786324951b0576f5d71fd820d6ef13dc/data.