People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Abu Bakr Belkaid University – Tlemcen
Faculty of Science and Technology / Department of Biomedical Engineering

FINAL YEAR PROJECT REPORT

In Partial Fulfillment of the Requirements for the Degree of MASTER in BIOMEDICAL ENGINEERING

Specialization: Medical imaging

Defended on: June 17, 2025

Managing Radiotherapy-Induced Side Effects with Artificial Intelligence: Early Detection of Muscle Atrophy and Personalized Nutritional Support in Head and Neck Cancer

---

Presented by:
DOUZI Houdallah

Before the Jury:

Mr. BOUKLI HACENE ISMAIL	*Professor*	**University of Tlemcen**	*President*
Mr. LAZOUNI Mohammed El Amine	*Senior lecturer (A)*	**University of Tlemcen**	*Examiner*
Mrs. KHEMIS Kamila	*Senior lecturer (A)*	**University of Tlemcen**	*Supervisor*
Mrs. BENHAMED Amina	*PhD Candidate in Medical Imaging*	**University of Tlemcen**	Co-Supervisor
Mrs. Douzi Meriem El Batoul	*PhD Candidate in nutrition and pathology*	**University of Tlemcen**	Co-Supervisor

Academic Year: 2025–2026

Summary

This project is based on a crucial step in medical image processing: segmentation. Using a trained U-Net model, we segmented two key muscles the masseter and the medial pterygoid located in the neck region of head and neck cancer patients following radiotherapy.
Thanks to satisfactory segmentation results, we were able to calculate the surface area of these muscles before and after treatment, allowing us to detect potential muscle atrophy in that region.

All of this was integrated into a simple, fast, and well-structured interface, designed for objective and efficient use.

The main goal of this project is to better manage the side effects of radiotherapy by offering personalized nutritional recommendations, improving patient follow-up and overall quality of life.
Key words: Medical image segmentation, U-Net model, Muscle atrophy detection, Head and neck cancer, Radiotherapy side effects, Personalized nutrition.

ملخص

يرتكز هذا المشروع على خطوة حاسمة في معالجة الصور الطبية، وهي التقطيع (التمييز). قمنا باستخدام نموذج U-Net المدرب لتقطيع عضلتين أساسيتين العضلة الماضغة والعضلة الجناحية الإنسيّة في منطقة الرقبة لدى مرضى سرطان الرأس والعنق بعد العلاج الإشعاعي.بفضل نتائج التقطيع المرضية، تمكّنا من حساب مساحة هذه العضلات قبل وبعد العلاج، مما أتاح لنا إمكانية الكشف عن الضمور العضلي في تلك المنطقة.

وقد تم تجميع كل هذه الخطوات داخل واجهة سهلة وسريعة وبنية واضحة لتوفير استخدام عملي وفعّال. الهدف الرئيسي من هذا المشروع هو تحسين التعامل مع الآثار الجانبية للعلاج الإشعاعي من خلال تقديم توصيات غذائية مخصصة، وتعزيز المتابعة الطبية وتحسين جودة حياة المرضى.

الكلمات المفتاحية :تقسيم الصور الطبية، نموذجU-Net ، كشف ضمور العضلات، سرطان الرأس والعنق، الآثار الجانبية للعلاج الإشعاعي، التغذية المخصصة.

Résumé

Ce projet repose sur une étape cruciale du traitement d'image : la segmentation. En s'appuyant sur l'entraînement d'un modèle U-Net, nous avons segmenté deux muscles clés le masseter et le ptérygoïdien médial dans la région cervicale de patients atteints de cancer ORL, après un traitement par radiothérapie.
Grâce à une segmentation satisfaisante, nous avons pu calculer la surface de ces muscles avant et après traitement, afin de détecter une éventuelle atrophie musculaire localisée.
L'ensemble de ce processus a été intégré dans une interface simple, rapide et bien structurée, permettant une manipulation fluide et objective.
L'objectif de ce projet est de mieux gérer les effets secondaires de la radiothérapie en proposant des recommandations nutritionnelles personnalisées, améliorer le suivi clinique des patients, et contribuer ainsi à une meilleure qualité de vie.
Mots-clés : Segmentation d'images médicales, Modèle U-Net, Détection de l'atrophie musculaire, Cancer de la tête et du cou, Effets secondaires de la radiothérapie, Nutrition personnalisée.

Acknowledgements

First and foremost, I offer my deepest and most sincere gratitude to **Almighty God**, the Most Merciful, the Most Compassionate. It is through His divine guidance, infinite mercy, and boundless grace that I have had the strength, patience, and perseverance to complete this work. Every step of this journey has been possible only by His will. In moments of doubt, He gave me clarity; in moments of fatigue, He granted me the strength to carry on. To Him belongs all praise, now and forever.

I wish to extend my heartfelt thanks to **Mrs. KHEMIS Kamila**, my thesis supervisor, for her valuable guidance, unwavering support, and constructive feedback throughout this project. Her availability, encouragement, and commitment have been instrumental to the completion of this research.

A special and sincere thank you to my co-supervisor, **Mrs. BENHAMED Amina**, for her precious help, constant support, and genuine dedication, which brought an essential added value to this work.

I would also like to express my deep appreciation to my sister, **Mrs. Douzi Meriem El Batoul**, who accompanied me in her role as a nutritionist, and generously provided me with insightful information that greatly enriched the quality of this study.

I would like to sincerely thank **the members of the jury** for the time they devoted to reading my thesis and for being part of my defense.

My gratitude also goes to all the professors of the **Master's program in Medical Imaging**, for the excellence of their teaching and their commitment to sharing knowledge with rigor and passion.

I am forever grateful to my **family and loved ones** for their unconditional love, constant encouragement, and moral support throughout my academic journey.

To everyone who, in one way or another, contributed to the completion of this thesis, I extend my sincere thanks.

Dedication

To **myself**,

For persevering through doubt, overcoming challenges, and believing in my abilities. This work is the result of my determination, my efforts, and my unwavering hope in the future. Despite the lack of resources and being rejected many times, I never gave up—I held on and made it to the end.

To **my parents**,

For their unconditional love, silent prayers, immense sacrifices, and constant support. Every time I needed help, they were there. Everything I ever wished for, they did their best to make it happen. None of this would have been possible without their presence, their patience, and their trust in me. This thesis belongs to them as much as it does to me.

List of Figures

Diagram III.1 A simplified schematic representation illustrating the interaction between the key components of a U-Net architecture, including the encoder, decoder, and skip connections.

Diagram III.2 Typical flow structure.

List of Tables

List of Abbreviations

HNC: Head and **N**eck **C**ancer
RT: Radiation **T**herapy
LAHNC: Locally **A**dvanced **H**ead and **N**eck **C**ancer
R/M HNC: Recurrent and/or **M**etastatic **H**ead and **N**eck **C**ancer
S: Surgery
CRT: Concurrent **R**adiotherapy and **C**hemotherapy
IC: Induction **C**hemotherapy
DNA: Deoxyribo**n**ucleic **A**cid
CT-PET: Computed **T**omography – **P**ositron **E**mission **T**omography
MRI: Magnetic **R**esonance **I**maging
CT: Computed **T**omography
AI: Artificial **I**ntelligence
ROI: Region **O**f **I**nterest
TSE: Turbo **S**pin **E**cho (an MRI sequence)
HU: Hounsfield **U**nit (measure of radiodensity in CT imaging)
ML: Machine **L**earning
CNN: Convolutional **N**eural **N**etwork
FCN: Fully **C**onvolutional **N**etwork
U-Net: U-shaped CNN architecture for biomedical image segmentation
ReLU: Rectified **L**inear **U**nit (activation function used in neural networks)
TCIA: The **C**ancer **I**maging **A**rchive
HNSCC: Head and **N**eck **S**quamous **C**ell **C**arcinoma
DICOM: Digital **I**maging and **C**ommunications in **M**edicine
MD: Medical **D**octor
NRRD: Nearly **R**aw **R**aster **D**ata (file format used for storing medical images)
BMI: Body **M**ass **I**ndex

Table of Contents

GENERAL INTRODUCTION

General introduction

Head and neck cancer refers to a group of malignant tumors that affect the anatomical structures located between the base of the skull and the clavicle. It primarily includes the oral cavity, pharynx, larynx, and salivary glands.

These cancers are prevalent worldwide and represent a major public health concern due to the complexity of the affected anatomy, their functional consequences, and the significant post-treatment complications they often entail.

Treatment usually involves a combination of surgery, chemotherapy, and most importantly, **radiotherapy**, a targeted therapeutic approach aimed at preserving the vital and aesthetic functions of the affected areas.

However, despite its clinical efficacy, radiotherapy frequently leads to significant side effects, among the most debilitating being **muscle atrophy** of the masticatory muscles, such as the **masseter** and **medial pterygoid**.

This progressive loss of muscle mass is a primary cause of **trismus**, a severe restriction in mouth opening that hinders chewing, swallowing, and verbal communication, ultimately compromising nutritional intake.

These complications greatly impair patients' quality of life, especially when muscle atrophy is diagnosed too late for effective intervention. In this context, **early detection** of muscular atrophy becomes a clinical priority to enable timely and targeted nutritional management.

Yet, the evaluation tools currently used in clinical practice remain largely **manual**, **time-consuming**, and **prone to human error**. They are therefore inadequate for large-scale, standardized, and rapid patient monitoring.

This is where **artificial intelligence** (AI) becomes a game changer. **Convolutional Neural Networks (CNNs)**, in particular, have revolutionized medical image analysis by offering powerful, automated methods for identifying and classifying complex anatomical structures. Remarkably, these technologies are inspired by the human brain, especially the visual cortex.

Throughout history, technological innovation has often drawn from both human physiology and the universe. It is this deep understanding of **cognitive processes**, **object recognition**, and **neural learning mechanisms** that led to the development of CNNs, models designed to detect, learn, and interpret shapes in images similarly to the way a developing brain perceives visual input.

Among these architectures, the **U-Net model** has emerged as particularly effective for biomedical image segmentation. With its ability to accurately identify anatomical contours from annotated datasets, it has become a cornerstone for quantitative medical analysis.

This project builds upon that potential, aiming to develop a system capable of **automatically segmenting masticatory muscles**, **estimating their surface area**, **detecting potential atrophy**, and **generating personalized nutritional recommendations**, all integrated within a user-friendly and interactive clinical interface.

Our work also reflects a broader **holistic vision of healthcare**, in which nutrition is not merely an adjunct to therapy, but a **core pillar** of treatment, prevention, and patient recovery. Contemporary

trends in **nutritional oncology** highlight the importance of individualized diets, rich in protein, natural antioxidants, and tailored to patients' oral and functional capacities, to improve clinical outcomes and maintain quality of life.

In this context, our tool offers not only **diagnostic support**, but also serves as a **bridge between biomedical analysis and clinical nutritional decision-making**.

Beyond its scientific contribution, this project also carries a profound **spiritual and human dimension**. The Holy Qur'an reminds us: ***"We have certainly created man in the best of stature"*** (Surah At-Tin, Verse 4), emphasizing the nobility of the human form and our duty to preserve its integrity. By identifying and preventing muscle degradation, restoring lost functions, and improving patient care, this work embodies a noble mission, one in which **science and compassion** unite to honor human creation through knowledge and service.

Guided by this vision, the project ultimately seeks to answer a crucial question in modern healthcare: **How can the potential of artificial intelligence, particularly U-Net-based automatic segmentation, be harnessed to detect post-radiotherapy muscular atrophy and thereby enhance personalized nutritional management for patients with head and neck cancer?**

To address this question, the research is structured to first explore the medical context of head and neck cancers, their treatment modalities, and the functional consequences of radiotherapy, especially regarding muscle atrophy and its early detection.

It then moves into the technical domain by reviewing deep learning approaches, with particular emphasis on CNNs and the justification for the choice of U-Net for muscle segmentation.

Finally, the work presents the practical implementation of an automatic segmentation system that estimates muscle surface areas and generates personalized nutritional recommendations, thereby bridging clinical needs and technological innovation.

Chapter I
MEDICAL CONTEXT

Chapter I: MEDICAL CONTEXT

I.1 Introduction

With the advancement of medical technologies, it has become increasingly possible to treat complex diseases such as cancer more effectively and precisely. Techniques like radiotherapy have become essential components in oncological care. However, like any intensive treatment, they come with their own side effects. These adverse effects can be particularly concerning when cancer is located in anatomically sensitive regions, such as the head and neck, where critical structures are densely packed and highly functional.

I.2 Definition of Head and Neck cancer

Head and neck cancer (HNC) ranks as the seventh most prevalent cancer worldwide, with over 660,000 new cases and approximately 325,000 deaths reported each year. [1]
Head and neck cancers typically arise from squamous cells lining the mucosal surfaces of the oral cavity, pharynx, and larynx, and are therefore commonly referred to as squamous cell carcinomas. Less frequently, tumors may also develop in the salivary glands, paranasal sinuses, or within the muscular and neural structures of the head and neck region. [2]

The most common anatomical sites include **(Figure I.1)**:

- **Oral cavity**
  Lips, anterior two-thirds of the tongue, gums, inner cheeks, floor of the mouth, hard palate, and the area behind the wisdom teeth.
- **Pharynx**
  Which is divided into the nasopharynx (behind the nose), oropharynx (soft palate, base of the tongue, tonsils), and hypopharynx (lower part of the pharynx).
- **Larynx**
  Located just below the pharynx, which houses the vocal cords and epiglottis.
  In rare regions:
- **Paranasal sinuses and nasal cavity**
- **Salivary glands**
  Mainly found on the floor of the mouth and near the jawbone, with additional minor glands throughout the oral and pharyngeal mucosa. [2]

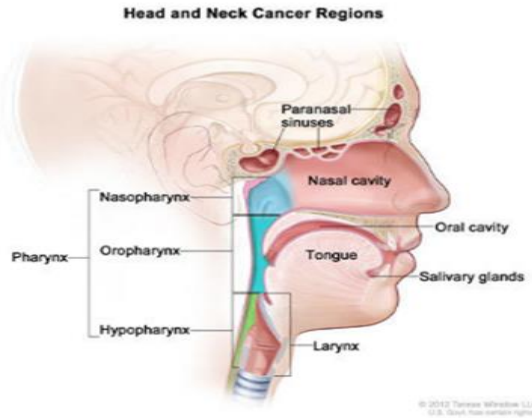


Figure I.1 Common anatomical sites of Head and Neck cancer. [3] [4]

I.3 Treatment of Head and Neck cancer

Head and neck cancer is often treated using a combination of methods, depending on the stage and tumor location, including chemotherapy, surgery, and radiotherapy **(Figure I.2)**.[5] Among these, radiation therapy (RT) plays a central role in curative-intent treatment strategies [6], significantly contributing to improved survival rates, functional outcomes, and quality of life. [7]

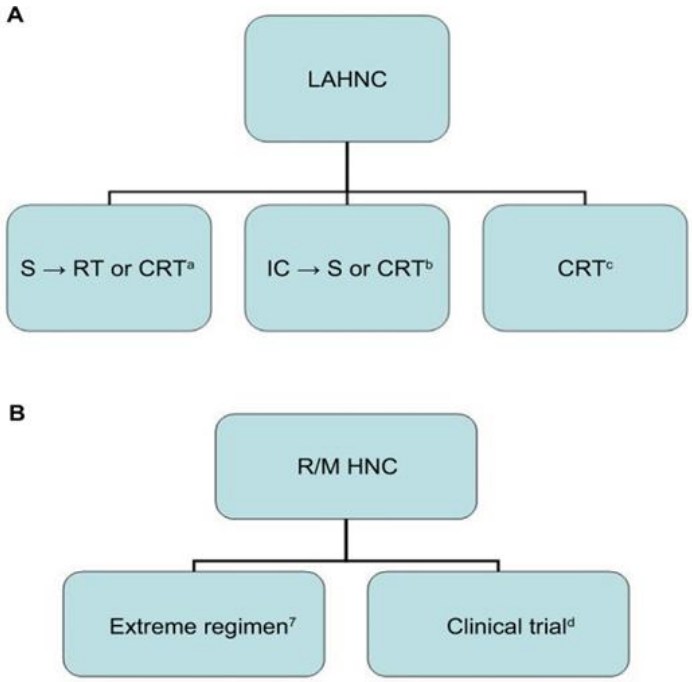


Figure I.2 Schematic flowchart suggested for the treatment of LAHNC (A) and R/M HNC (B). [7]

I.4 Radiotherapy and secondary effects side

Radiotherapy is an oncological treatment that relies on the use of ionizing radiation to eliminate cancer cells **(Figure I.3)**. [8]

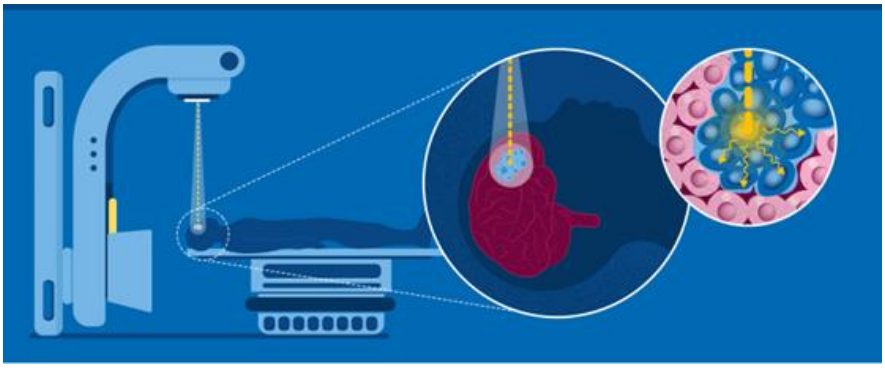


Figure I.3 External radiotherapy. [9]

Curative radiotherapy for head and neck tumors delivers high doses to areas near critical structures. [6] Doses range from 54 to 70 Gy, delivered using a standard fractionation schedule of 2 Gy per fraction, one fraction per day, five days a week. [10]

External beam radiotherapy is widely used in the management of locally advanced head and neck cancers. It works by delivering ionizing radiation to induce irreparable DNA damage in cancer cells. However, adjacent healthy structures, especially muscles and nerves, also receive significant radiation doses, leading to acute and late toxicities **(Figure I.4)**. [11]

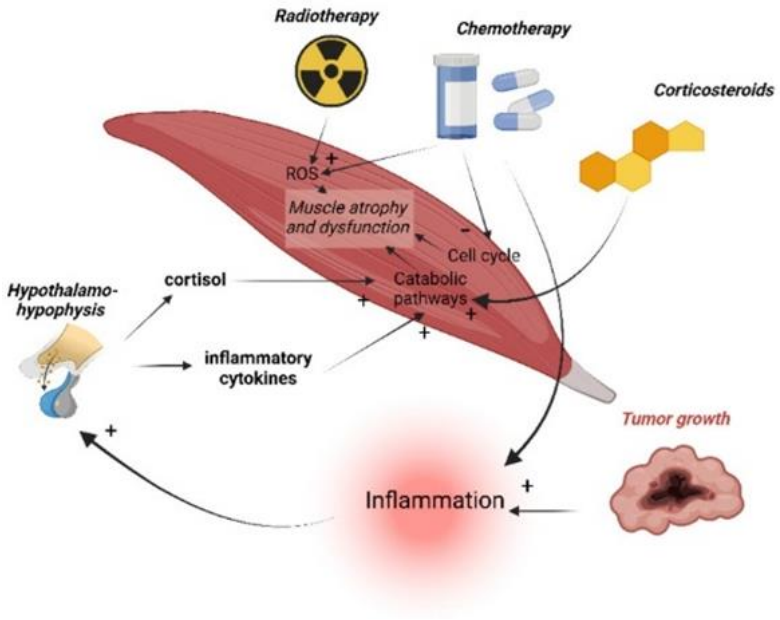


Figure I.4 Schematic of the muscle direct- or indirect-related toxicities (muscle atrophy and dysfunc tion) induced by cancer and anti-cancer treatments (radiotherapy, chemotherapy, corticosteroids). [11]

Radiotherapy increases the production of **reactive oxygen species (ROS)** in muscle cells. These ROS cause cellular damage, which activates **inflammatory** and **catabolic pathways**, leading to **protein degradation**. As a result, patients experience **muscle atrophy** and functional decline, which can worsen during treatment.

I.4.1 Muscle atrophy post-RT

Neck muscle atrophy and soft-tissue fibrosis are recognized late complications in head and neck cancer survivors. [12]

Skeletal muscle atrophy **(Figure I.5)**, which is marked by a reduction in muscle strength and mass, and can profoundly diminish a patient's quality of life. [13] [14]

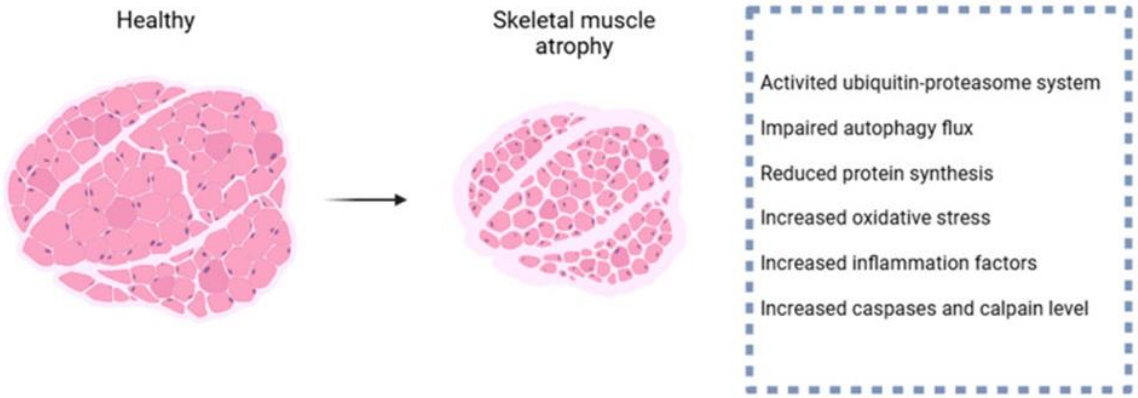


Figure I.5 The pathogenesis of skeletal muscle atrophy. [13]

In particular, in head and neck cancer the repeated exposure of masticatory muscles to radiation can result in chronic inflammation, fibrosis, and progressive muscular degeneration. These effects manifest clinically as muscle atrophy, often affecting **the masseter** and **pterygoid muscles**, which play key roles in jaw movement and chewing. This atrophy contributes significantly to the onset of trismus, which limits the ability to open the mouth adequately. [15] [16]

Muscle area measurements showed excellent reliability (>96%), and after a median 56 Gy dose and 2.9 years' follow-up, the masseter's cross-sectional area declined by 0.17 cm² per year (3.9%/y) and the medial pterygoid's by 0.13 cm² per year (2.3%/y) **(Figure I.6) (Table I.1)**. [17]

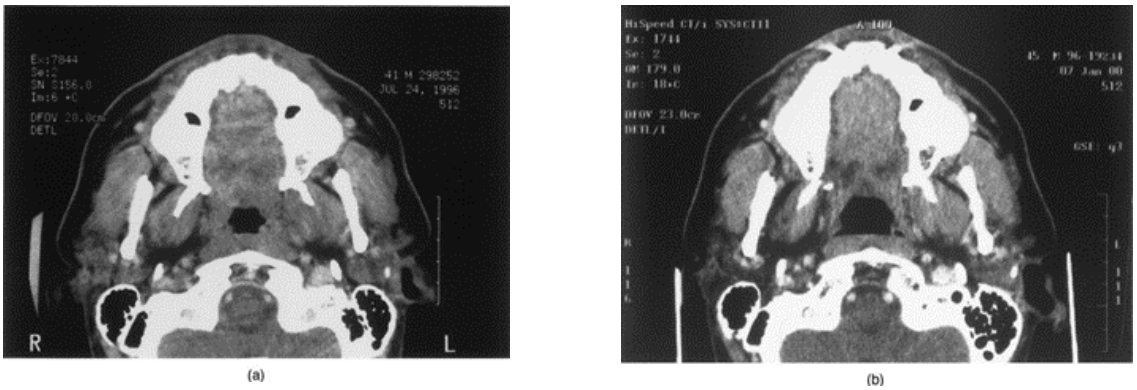


Figure I.6 Axial CT scans in the reference plane (a) pretreatment (b) 3.4 years after 66 Gy in 33 fractions to the right side with a wedge pair technique showing 11% masseter atrophy and 28% medial pterygoid atrophy. [17]

Muscle	No atrophy (Grade 0)	≤10% (Grade 1)	>10–20% (Grade 2)	>20–50% (Grade 3)	>50% (Grade 4)
Masseter	2	6	1	4	0
Medial pterygoid	4	4	2	4	0
Masseter or medial pterygoid	0	5	2	7	0

Table I.1 Data presented as the number of patients. Abbreviation: LENT-SOMA = Late Effects Normal Tissues–Subjective, Objective, Management, Analytic. [17]

Masseter muscle

The masseter muscle is a key component of the masticatory system, playing an essential role in jaw movement and chewing function **(Figure I.7)**. [18] Previous studies suggest that alterations in the morphology of the masseter muscle may contribute to structural changes in the buccal region of the face. [19]

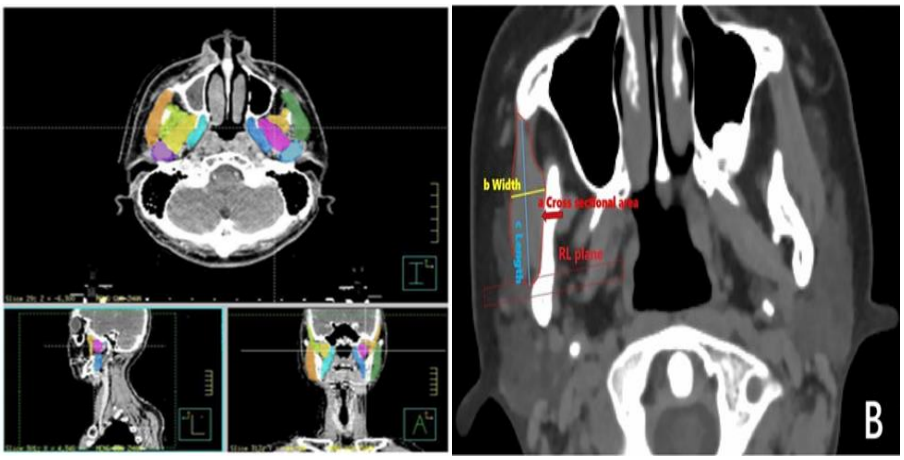


Figure I.7 Describe the trismus related muscles and structures that were targeted by colors in one of enrolled patients. Light blue, Sky blue: medial pterygoid muscle; green, Orange: masseter muscle. [19]

Medial pterygoid muscle

The medial pterygoid, also known as the internal pterygoid muscle (**Figure I.8**), is a square-shaped muscle involved in mastication. Positioned on the inner side of the lower jaw on both sides, it serves as a key elevator of the mandible and lies medial to the lateral pterygoid muscle. [20]

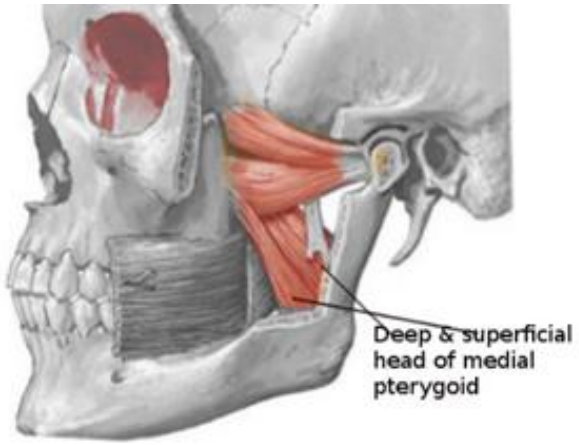


Figure I.8 Medial pterygoid muscle Image. [20]

I.4.2 Trismus

Trismus is defined as a maximum interincisal opening of less than 35 mm. It is caused by neuromuscular complications such as musculoskeletal pain, spasm, tissue fibrosis, and muscle weakness **(Figure I.9)**. In RT, the structures associated with radiation-induced trismus until now have been identified as the masseter muscles, pterygoid muscles, and temporomandibular joints. [21] [22]

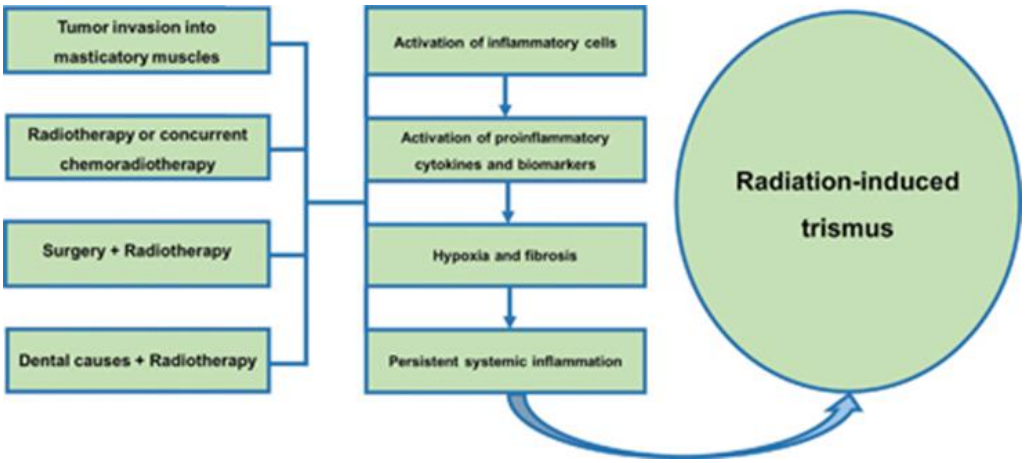


Figure I.9 Summary of the etiology and pathogenesis of radiation-induced trismus. [22]

Side effects	Symptoms	Etiology	Complications	Management
Trismus	Mastication muscles fibrosis	inability to open the mouth	Adverse effects on oral care, chewing, nutrition, oral care, speech production	Oral appliances, diet modification, pharyngeal strengthening, swallow retraining

Table I.2 Late radiation toxicities of head and neck cancers. [23]

In this case patients suffering from trismus (moderate to severe trismus) can significantly impact patient health and quality of life by impairing chewing, nutrition, and access to the oral cavity **(Table I.2)**. [24]

I.5 Nutrition in managing the muscle atrophy

The management of radiotherapy side effects remains a recurrent challenge for radiation oncologists, and various treatment options are proposed depending on specific clinical contexts. [25] As previously discussed, managing muscle atrophy can help prevent the onset of trismus, thereby improving both the effectiveness of treatment and patients' quality of life.

Understanding the molecular mechanisms underlying skeletal muscle atrophy and applying appropriate therapeutic strategies can significantly aid muscle function recovery and enhance patients' overall well-being **(Figure I.10) (Figure I.11)**. [26]

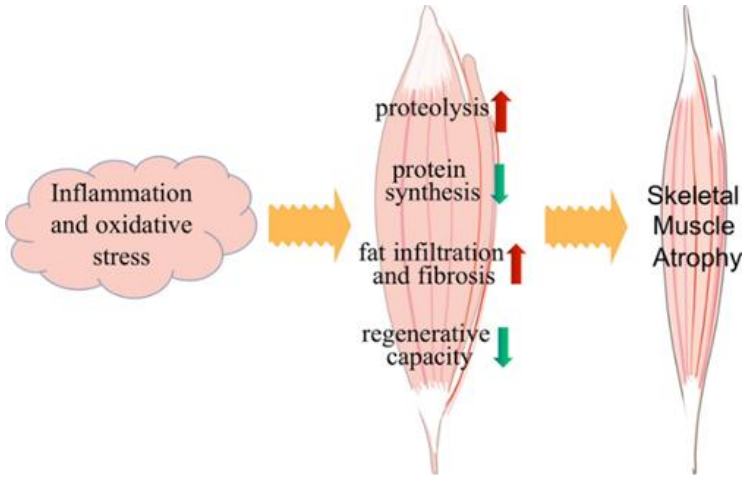


Figure I.10 Molecular mechanism of skeletal muscle atrophy. Inflammation and oxidative stress lead to increased proteolysis, reduced protein synthesis, decreased regenerative capacity and increased fat infiltration and fibrosis. [27]

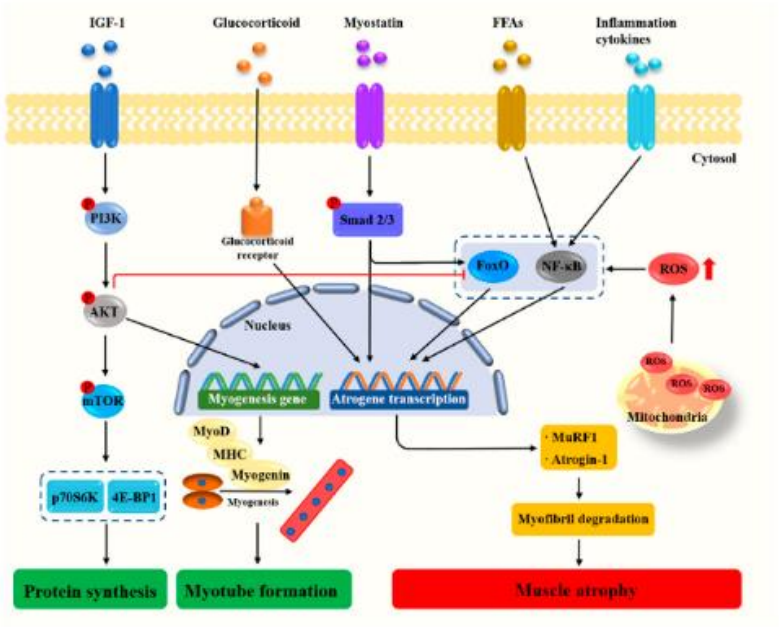


Figure I.11 Cellular and molecular mechanisms regulating muscle growth and atrophy. [28]

Skeletal muscle comprises approximately 40%–50% of body mass and plays a critical role in human energy metabolism. Its structural and functional integrity is essential for maintaining musculoskeletal and metabolic health.

This has made the search for dietary strategies to prevent or treat muscle atrophy and sarcopenia a prominent focus in nutrition, medicine, and biology **(Figure I.12)**.

Oxidative stress and inflammation are key contributors to skeletal muscle degradation. Reducing dietary intake of pro-inflammatory components may therefore help mitigate sarcopenia.

Muscle atrophy can be exacerbated by several dietary deficiencies, such as inadequate protein, insufficient vitamin D, and a lack of antioxidants and long-chain polyunsaturated fatty acids.

In men, handgrip strength has been significantly associated with the composite dietary antioxidant index and intake of zinc, vitamin E, and selenium, while in women, only zinc showed a significant correlation.

Hence, maintaining a varied and balanced diet is essential for preserving physiological function. [28]

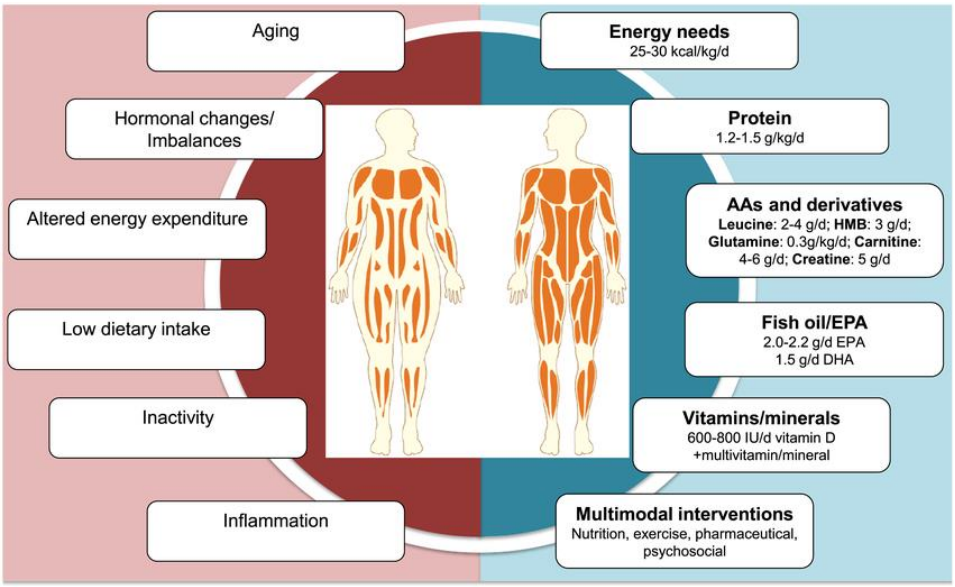


Figure I.12 Causes and potential nutrition treatments for low muscle mass in cancer. [29]

Nutrition plays a central role in health maintenance. While radiotherapy (RT) remains crucial in cancer management, one of its adverse effects **"trismus"** results from atrophy of the masseter and medial pterygoid muscles, which are vital for mastication. This complication not only diminishes quality of life but also interferes with treatment efficacy due to compromised nutritional intake.

Growing evidence highlights the benefits of integrating physical activity and nutritional strategies during and after cancer therapy. These interventions are now recommended as part of standard care. Nutritional support is vital to prevent and manage malnutrition, sarcopenia, and cachexia. When combined with tailored physical activity, these measures help alleviate the symptom burden of both the disease and its treatments. [30]

I.6 Conclusion

This chapter provided a comprehensive overview of the medical context surrounding head and neck cancer. We began by defining the disease and outlining the common treatment modalities, with a particular focus on radiotherapy. While radiotherapy remains a cornerstone of cancer management, it is often accompanied by significant side effects, including muscle atrophy and trismus, which can severely impact patients' quality of life. Special attention was given to the masseter and pterygoid muscles due to their critical role in mastication and speech. Additionally, we explored the role of nutrition in mitigating muscle degradation and supporting patient recovery. Understanding these medical challenges is essential for developing effective image analysis tools and personalized interventions, as explored in the subsequent chapters.

Chapter II

MUSCLE SEGMENTATION IN MEDICAL IMAGING

Chapter II: MUSCLE SEGMENTATION IN MEDICAL IMAGING

II.1 Role of medical imaging in managing the side effects of RT

Medical imaging is a discipline that has enabled healthcare professionals—and even developers—to explore the human body with remarkable precision and high resolution. Beyond that, it has paved the way for new avenues of research and innovation. Medical imaging technologies have significantly advanced the diagnosis, management, and prediction of various diseases through different modalities such as:

- CT-PET
- MRI
- CT scan

Which have become essential tools in modern medical research.

II.1.1 CT SCAN images

CT scan imaging **(Figure II.1)** offers objective, quantitative, and qualitative assessment of skeletal muscle and adipose tissue. Owing to its routine use in clinical settings, CT is a readily available tool for monitoring body composition changes during treatment, making it a valuable imaging biomarker for predicting clinical outcomes. [31]

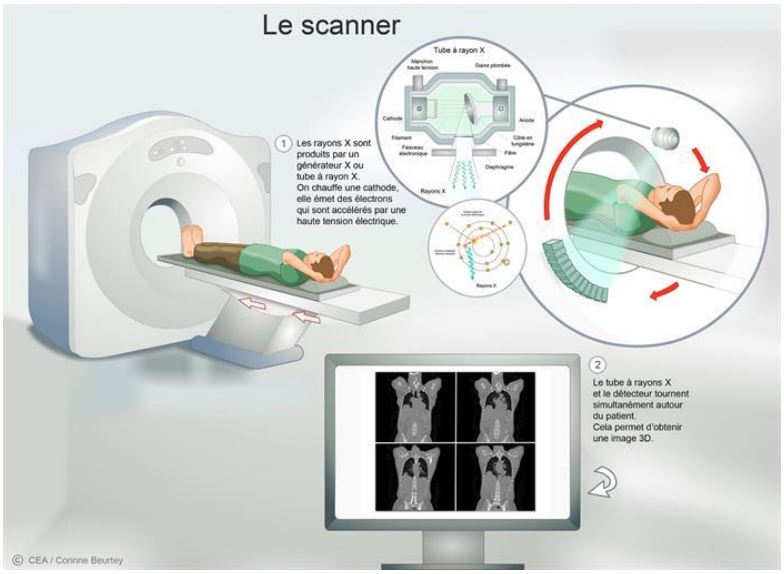


Figure II.1 CT scan. [32]

In our study, we aim to evaluate muscle atrophy in the masseter and pterygoid muscles following radiotherapy (RT) using axial CT images **(Figure II.2)** . To achieve this, we calculate the muscle surface areas for comparison. This approach requires performing muscle segmentation to accurately extract the targeted muscles.

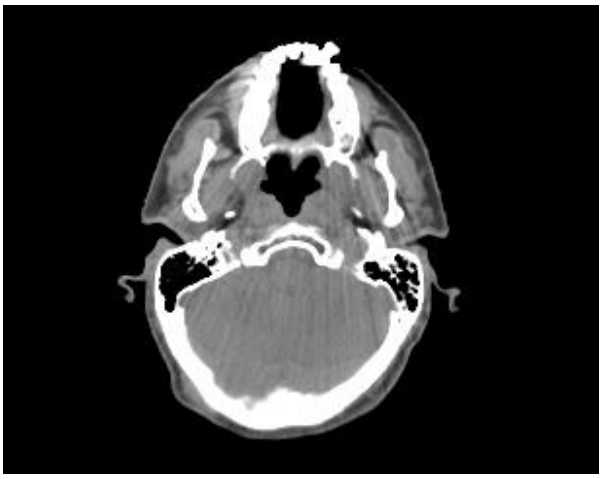


Figure II.2 Axial CT image of the neck was used for comparison. [33]

II.2 Muscular segmentation

II.2.1 Definition of muscular segmentation

Segmentation of medical images is a crucial step for accurate diagnosis and effective treatment planning, enabling detailed and precise analysis across various clinical applications. [34] It can be defined as a method for isolating specific structures within medical images. This process may be performed manually, semi-automatically, or fully automatically. In the context of head and neck cancers, our focus is on segmenting the masseter and medial pterygoid muscles to detect the side effects of radiotherapy (RT), identify signs of muscle atrophy, and correlate these findings with functional impairments such as trismus.

II.2.2 Methods of segmentation

II.2.2.1 Manual methods

Manual muscle segmentation involves the process of manually outlining muscle structures in medical images (typically MRI or CT scans) by an expert, such as a radiologist or anatomist **(Figure II.3)**. It is considered the gold standard in terms of precision, especially when training data for AI or validating automatic segmentation algorithms. [35]

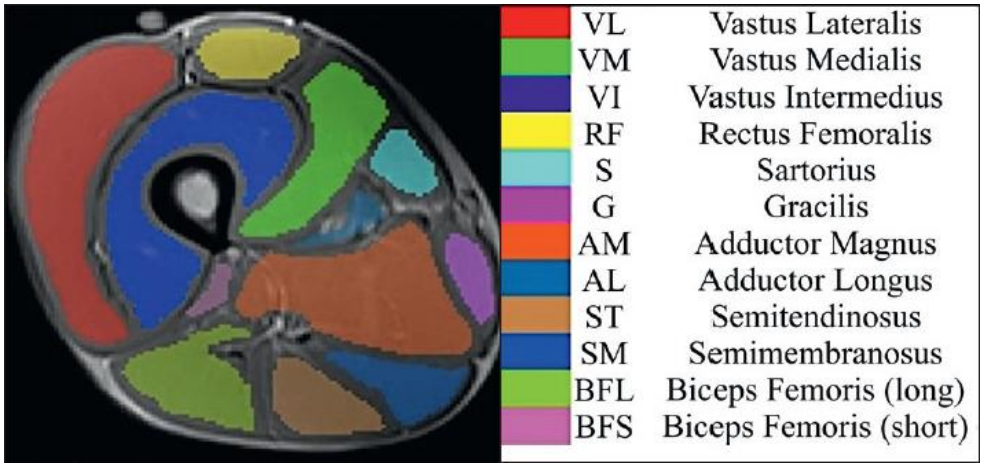


Figure II.3 Example of ROIs obtained during manual thigh muscle segmentation superimposed on a TSE T2 weighted image. [35]

- **Advantages**
- High accuracy and anatomical reliability
- Preferred for ground truth generation in research
- Useful in complex or noisy datasets where automated models fail
- **Limitations**
- Time-consuming and labor-intensive
- Inter-observer variability (results can vary depending on the operator)
- Not scalable for large datasets or clinical workflows. [36]
- **Tools Commonly Used**
- ITK-SNAP
- 3D Slicer
- MITK

These tools allow manual delineation slice-by-slice in 3D images.

II.2.2.2 Semi-automatic methods

In medical image analysis, algorithmic techniques such as thresholding and mathematical morphology play a key role in semi-automatic segmentation of muscle tissue **(Figure II.4)**. These methods are essential for improving the precision and reproducibility of muscle delineation, especially in anatomically complex regions.

❖ Thresholding Techniques

Thresholding is a fundamental image segmentation technique that classifies pixels based on their intensity values to isolate muscle structures from surrounding tissues.

- **Global thresholding** applies a single intensity cutoff across the image, which is effective when muscle contrast is consistent.
- **Adaptive thresholding**, in contrast, adjusts the threshold locally, enabling better segmentation in images with uneven illumination or variable tissue density.

Despite their simplicity and speed, thresholding methods can be limited by image noise, overlapping tissue intensities, and poor contrast. [37]

❖ Mathematical Morphology

Mathematical morphology provides a suite of pixel-based operations designed to refine binary segmentation results. These operations manipulate the shape and connectivity of segmented regions to enhance anatomical accuracy.

- **Erosion** removes small, isolated regions (often noise).
- **Dilation** expands segmented regions to restore structure lost during erosion.
- **Opening** (erosion followed by dilation) smooths object contours and removes narrow protrusions.
- **Closing** (dilation followed by erosion) fills small gaps and holes within objects.

These morphological operations are critical in cleaning up segmentation masks, separating adjacent muscles, and refining edges, especially when initial thresholding is imperfect. [38]

❖ Combined Approach for Accurate Segmentation

The integration of thresholding and morphological operations forms an effective strategy for semi-automatic muscle segmentation. Thresholding provides a rapid initial segmentation, while

morphological refinement ensures anatomical plausibility and robustness. This combined approach enhances the reliability of muscle analysis, supporting both clinical decision-making and quantitative research applications.

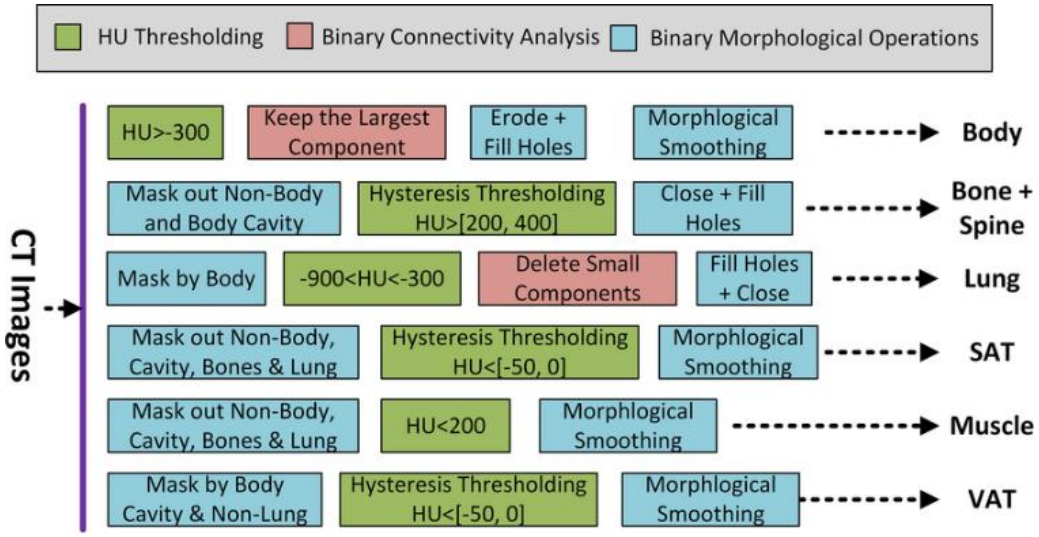


Figure II.4 Detailed procedures for HU thresholding and morphological operations in step 2. [39]

II.2.2.3 Automatic methods

Automatic segmentation methods aim to extract muscle structures from medical images without human intervention. These approaches leverage algorithms that can learn patterns, detect anatomical structures, and delineate tissue boundaries based on image features such as intensity, texture, and shape. Automatic methods are especially valuable for handling large-scale datasets and reducing inter-operator variability in clinical analysis.

More recently, machine learning (ML) algorithms have been introduced to deal with the huge variability of clinical data. Specifically, Convolutional Neural Networks (CNNs) are emerging as the new state of the art for medical-image segmentation. One of the advantages of the CNNs is the ability to learn more robust features through a hierarchical structure composed of learnable filters, without relying on hand-crafted features. [40]

❖ CNNs Convolutional Neural Networks

In biomedical imaging, Convolutional Neural Networks (CNNs) are extensively applied for tasks such as automatic disease detection and classification, therapeutic outcome prediction, and the segmentation and identification of specific tissues and organs. These networks are trained and validated using annotated medical data provided by expert physicians. Once trained, CNNs can extract relevant features from new medical images, enabling accurate prediction and segmentation based on the learned representations **(Figure II.5)**. [41]

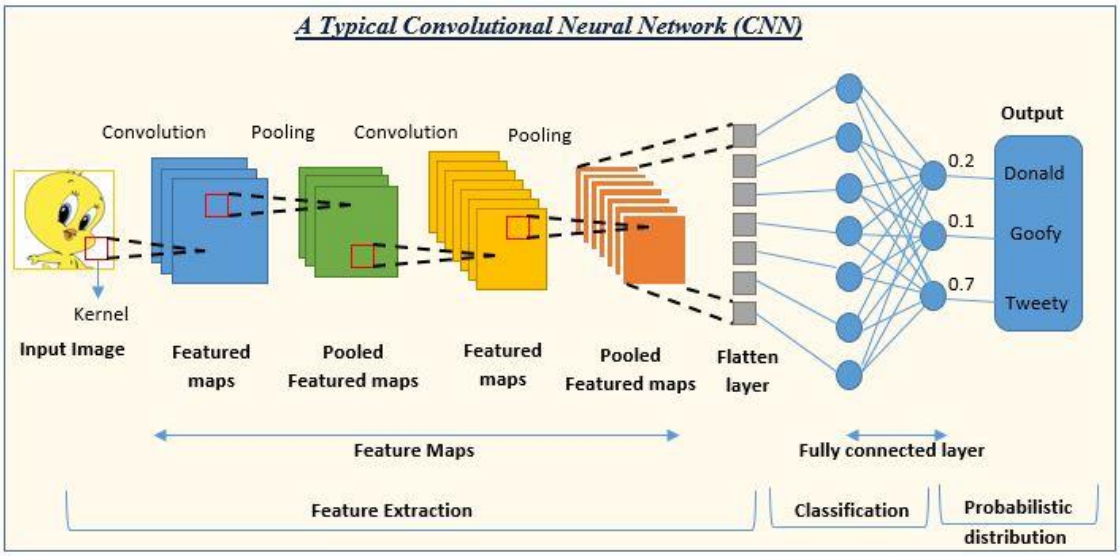


Figure II.5 Overview of Convolutional Neural Network. [41]

❖ Types of CNNs

- Basic CNN (Standard CNN) Deep CNN
- ResNet (Residual Networks)
- Inception Networks
- MobileNet / EfficientNet
- Fully Convolutional Networks (FCNs)
- U-Net
- R-CNN and Variants (Fast R-CNN, Faster R-CNN, Mask R-CNN)
- 3D CNN

To segment these muscles, we chose to use **U-Net** due to its **efficiency and reliability in medical image segmentation tasks**.

❖ U-net U network

U-Net is a deep learning architecture widely used for processing continuous signals, particularly in applications such as image segmentation, PDE surrogate modelling, and diffusion models. Initially proposed in 2015 by Ronneberger et al., it was specifically designed to address segmentation tasks with limited annotated data. The network performs pixel-wise classification, making it particularly suitable for accurately segmenting anatomical structures in medical images.

U-Net follows an **encoder–decoder** structure enhanced with **skip connections**, which help retain spatial information lost during downsampling and improve prediction accuracy. Its strong performance and flexibility in incorporating domain-specific knowledge make it especially effective in medical image analysis and other scientific fields requiring detailed spatial understanding **(Figure II.6)**. [42]

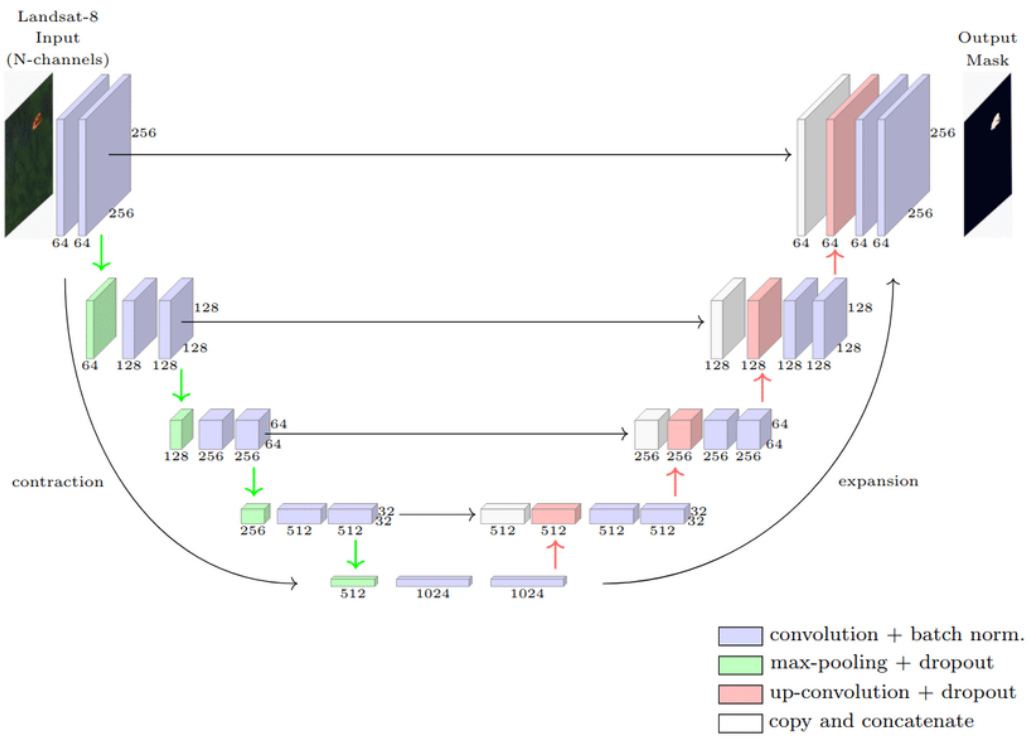


Figure II.6 Active fire recognition neural network. [43]

❖ Blocks of U-Net Architecture

1. **Encoder-Decoder Structure**
   - The U-Net is built around a symmetric encoder-decoder design.
2. **Skip Connections**
   - Connections between corresponding layers of the encoder and decoder to retain spatial information.
3. **Convolutional Blocks**
   - Typically include two convolutional layers followed by an activation function (e.g., ReLU). [44]

II.3 Conclusion

In this chapter, we explored the crucial role of medical imaging in evaluating and managing the side effects of radiotherapy, with a particular focus on muscle atrophy. Computed tomography (CT) emerged as a key imaging modality for identifying muscular changes, especially in head and neck cancer patients undergoing RT.

We then examined the concept of muscular segmentation, emphasizing its importance for accurate assessment and follow-up. Various segmentation methods were detailed, ranging from manual techniques to semi-automatic and fully automatic approaches. Manual and semi-automatic methods, though still widely used, often face challenges related to time and reproducibility.

With the advancement of artificial intelligence, automatic methods—especially those based on Convolutional Neural Networks (CNNs)—have gained prominence. Among them, the U-Net architecture stands out for its effectiveness and precision in medical image segmentation.

Understanding the structure and functioning of U-Net provides a solid foundation for its application in clinical research and practice.

This foundation sets the stage for the next chapter, which focuses on the practical implementation of U-Net for segmenting head and neck muscles, and its role in analysing treatment-related changes.

CHAPTER III

U-Net BASED SEGMENTATION IN HEAD NECK REGION IMAGES

CHAPTER III: U-Net-Based Segmentation of Head and Neck Region Images

III.1 Introduction

To meet the goal of this study, segmenting specific muscles before and after radiotherapy to assess muscle atrophy and help reduce its impact through personalized nutritional advice, we had to build a practical and well-organized approach from the ground up.

We started by putting together a carefully selected patient database, making sure that the CT scans were of good quality and clearly showed the relevant areas both before and after radiotherapy. Since our work focused specifically on the masseter and medial pterygoid muscles, we took time to accurately choose the CT slices where these muscles are best visible.

To improve the model's performance and make it more reliable, we applied preprocessing techniques and data augmentation. We then trained a U-Net model, widely used for medical image segmentation, so it could learn to recognize the two muscles effectively.

Once the model was trained, we used it to automatically segment the muscles, calculate their surface areas before and after treatment, and detect any muscle loss. Based on these results, we added a system that could generate personalized nutritional recommendations.

Finally, all of this was integrated into a complete, easy-to-use interface, allowing healthcare professionals to go from CT image analysis to clinical decision-making in just a few steps.

This chapter walks through each part of the implementation, from how the data was prepared to how the interface works, including the results and their importance in clinical practice.

III.2 Materials and Methods

III.2.1 Software tools

1) Python

Python is a versatile, high-level programming language widely used in fields like web development, AI, and data science. Its clear syntax enables rapid development of complex applications. Key libraries used in this project include:

- **SimpleITK** for medical image processing and handling DICOM/NIfTI files.
- **NumPy** for efficient numerical operations on arrays and matrices.
- **Matplotlib** for visualizing images and data.
- **OS** for file and system management.
- **OpenCV (cv2)** for advanced image processing tasks.
- **random** to introduce variability in data augmentation.
- **TensorFlow** and **Keras** to build and train neural networks efficiently.

2) MicroDicom

MicroDicom is a free DICOM viewer used to inspect medical images, perform measurements, annotate, and export images. It was helpful during pre-processing to verify image quality and content.

3) 3D Slicer

3D Slicer is an open-source platform offering advanced tools for 3D visualization, segmentation, and registration of medical images. It's widely used in research and clinical practice and helped validate segmentations and explore anatomical structures in this study.

4) React, CSS, and JavaScript

The user interface was built using **React**, a JavaScript library for creating dynamic, component-based web apps. **CSS** was used to style and animate the layout, while **JavaScript** handled interactivity and communication with the backend for image analysis and results display.

III.2.2 Dataset

- **Head-Neck-CT-Atlas**

The *Head and Neck Cancer CT Atlas* dataset is a publicly available subset of the **HNSCC** collection on **TCIA**, focused on patients diagnosed with head and neck squamous cell carcinoma (**HNSCC**) who underwent radiotherapy (**RT**). This dataset highlights the crucial role of cross-sectional imaging in the planning and monitoring of RT and addresses the general scarcity of publicly shared RT data, often due to the complexity of imaging structures and challenges in temporal and spatial registration. Out of **2,840** patients treated with curative-intent RT at MD Anderson Cancer Center between **2003** and **2013**, **215** were selected based on the availability of both **pre- and post-treatment PET-CT or abdominal CT scans**. Clinical information was retrieved from the institutional medical record system, and skeletal muscle and adipose tissue volumes were quantified from the imaging. All data were de-identified, reviewed, and uploaded to TCIA after quality control. The resulting dataset includes **433,384 DICOM** files across **3,225 series** and **765 studies**, providing matched clinical and imaging data—such as demographics, tumor grade, stage, recurrence, and survival outcomes. This open-access dataset supports multi-institutional research by enabling detailed analysis of RT planning, effectiveness, and associated toxicities. [45]

The dataset provides an Excel file containing comprehensive clinical and treatment information for each patient, including demographics, cancer staging, survival data, treatment modalities (surgery, chemotherapy, radiotherapy), smoking history, nutritional support details, and precise body composition measurements derived from CT images taken before and after radiotherapy.

❖ Dataset preparation

a. Data Preprocessing

From this database, a rigorous selection of patients was necessary. Many of the included patients had undergone a complete treatment protocol that involved not only RT but also surgery and chemotherapy **(Figure III.1)**.

Using the Excel file provided with the dataset, we filtered and selected only those patients who received RT alone. Out of the initial 215 patients, 100 were retained after this first step. Using a DICOM viewer such as MicroDicom **(Figure III.2)**, we reviewed the medical images of these

patients. However, not all of them had CT images; some only had PET scans, and others had poor image quality. As a result, the number of suitable patients was reduced to 50, each with clearly defined CT images both before and after RT. To reduce the complexity of the study, we selected a single axial slice that clearly shows both the masseter and medial pterygoid muscles. For each patient, one image before and one image after RT were chosen, resulting in two images per patient.

	TCIA PatientID	ic Survival Censor	Loco-regional Control Censor	Oncologic Treatment Summary	Induction Chemotherapy
2	HNSCC-01-0001	1	1	CCRT	No
3	HNSCC-01-0002	0	0	CCRT	No
4	HNSCC-01-0003	0	1	CCRT	No
5	HNSCC-01-0004	1	1	CCRT	No
6	HNSCC-01-0005	1	0	ERT	No
7	HNSCC-01-0006	0	1	CCRT	No
8	HNSCC-01-0007	1	1	Induction CMT --> CCRT --> Surgery	Carbo + Taxol x 2 cycles
9	HNSCC-01-0008	1	1	ERT	No
10	HNSCC-01-0009	1	1	Induction CMT --> ERT --> Surgery	Carboplatin + Taxol + Ifosfamide x 3 cycles
11	HNSCC-01-0010	1	0	CCRT	No
12	HNSCC-01-0011	1	1	Induction CMT --> CCRT --> Surgery	Carbo + Taxol x 2 cycles
13	HNSCC-01-0012	1	1	ERT --> Surgery	No
14	HNSCC-01-0013	1	1	ERT	No
15	HNSCC-01-0014	1	1	Surgery --> CCRT	No
16	HNSCC-01-0015	1	1	Induction CMT --> ERT	Carbo + Taxol x 4 cycles
17	HNSCC-01-0016	1	1	ERT --> Surgery	No

Figure III.1 Head-Neck-CT-Atlas Clinical Data [45]

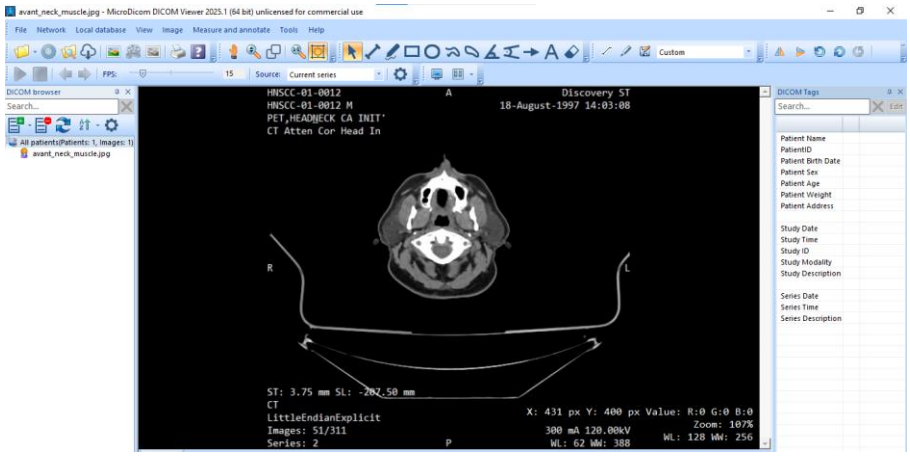
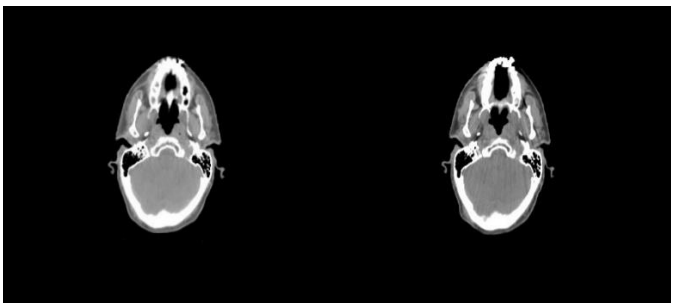


Figure III.2 MicroDicom Viewer Interface

We also removed the annotations and embedded metadata from the axial slices using a simple Python script **(Appendix 1)**. Additionally, basic preprocessing steps such as normalization, filtering, and contrast enhancement were applied to improve the overall image quality.

- **Illustrative patient sample: HNSCC-01-0093**



(A) (B)

Figure III.3 Representative axial neck CT images from one of the selected patients: (**A**) prior to radiotherapy, (**B**) following radiotherapy.

At this stage, each of the 50 selected patients has two high-quality axial CT images—one acquired before and the other during or after radiotherapy. All patients received RT exclusively, with no history of prior treatments. The images are free from artifacts such as annotations or overlaid text **(Figure III.3)**.

b. Muscle segmentation

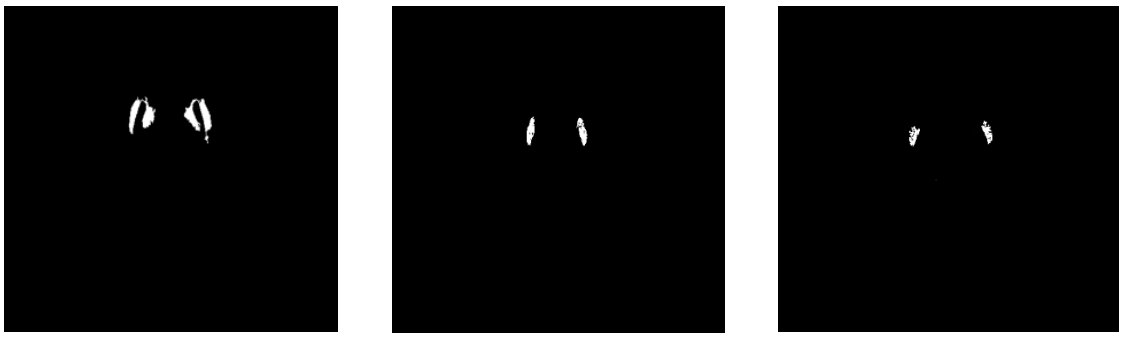
➢ Adaptative thresholding

Initially, we attempted a simpler method: **adaptive thresholding**. However, this approach did not yield consistent results across the dataset. In some images, only the masseter muscle was segmented, in others both muscles, and in a few, only the medial pterygoid **(Table III.1) (Figure III.4)**.

This inconsistency was primarily due to the similarity in gray levels and contrast between different anatomical structures, making it difficult to distinguish the muscles clearly. Additionally, some images did not provide a clear view of the muscles, and since the dataset was relatively old, image quality was often a limiting factor. Moreover, the shape and integrity of the segmented muscles were significantly affected by variations in threshold values, leading to distortions **(Figure III.5)**.

Number/type	Masseter Muscle	Medial pterygoid muscle	Both muscles
Number of patients	17	5	16

Table III.1 Table showing the number of patients segmented for each muscle using adaptive thresholding.



(1) (2) (3)

Figure III.4 Muscle segmentation using adaptive thresholding in three distinct scenarios: (1) simultaneous segmentation of both muscles, (2) segmentation of the masseter muscle only, (3) segmentation of the medial pterygoid muscle only.

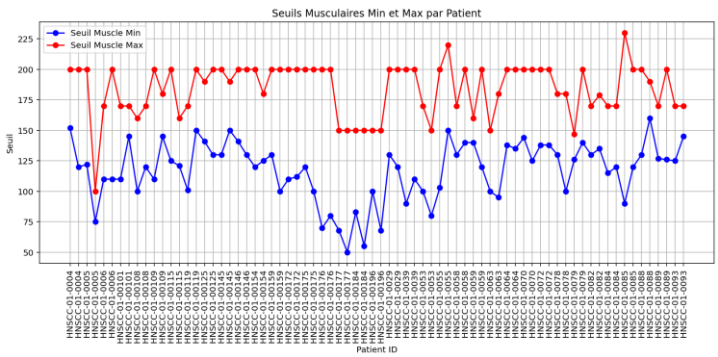


Figure III.5 Distribution of Muscle Intensity Thresholds (Min/Max) Across Patients.

These challenges made it necessary to move toward a more advanced and suitable segmentation approach.

- **Manual labeling for deep learning model training**

After carefully selecting our patients, the next essential step involved performing manual segmentation (also referred to as annotation) of the relevant muscles. This manual annotation serves as the foundation for training the segmentation model, providing precise ground truth data.

To achieve this, we used **3D Slicer**, we created **multiclass segmentation masks**, where each class represents a different anatomical structure. In our case, we defined **three classes**:

- **0** for the background,
- **1** for the **masseter muscle**,
- **2** for the **medial pterygoid muscle (Figure III.6)**.

Each class was associated with a distinct color to visually distinguish the two muscles. A **multiclass mask** refers to a segmentation map in which each pixel or voxel is assigned a specific integer label corresponding to a particular class. This allows the model to learn to differentiate between multiple anatomical structures simultaneously. [46]

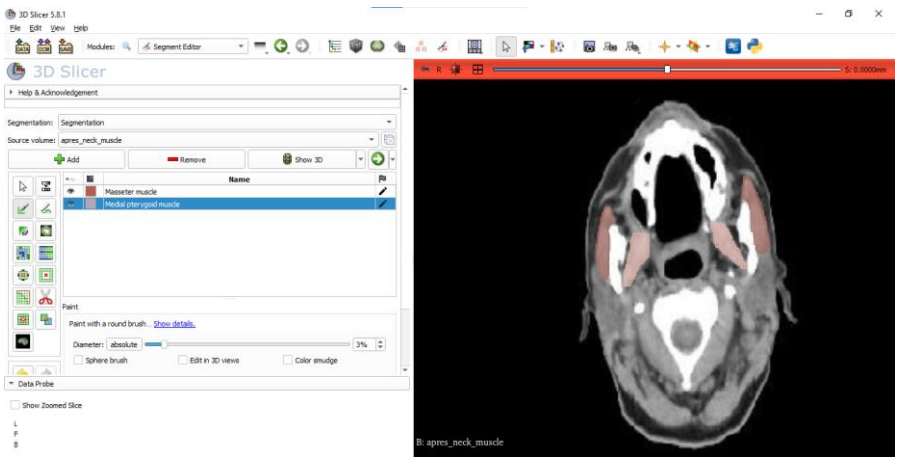


Figure III.6 Manual segmentation of the masseter and medial pterygoid muscles using the Paint tool in 3D Slicer, with distinct colors assigned to each muscle.

The masks were saved in the **.nrrd (Nearly Raw Raster Data)** format **(Figure III.7)**, which is particularly suited for medical imaging due to its support for multidimensional data and ease of integration with software like 3D Slicer. This format preserves spatial metadata (such as pixel spacing and orientation), which is crucial for maintaining anatomical accuracy across the dataset. [47]

```
NRRD0004
# Complete NRRD file format specification at:
# http://teem.sourceforge.net/nrrd/format.html
type: unsigned char
dimension: 3
space: left-posterior-superior
sizes: 512 512 1
space directions: (0.26458333333333339,0,0) (0,0.26458333333333339,0) (0,0,1)
kinds: domain domain domain
encoding: gzip
space origin: (0,0,0)
Segment0_Color:=0.752941 0.407843 0.345098
Segment0_ColorAutoGenerated:=1
Segment0_Extent:=180 311 184 229 0 0
Segment0_ID:=2.25.106932571764399150681239763074816604002
Segment0_LabelValue:=1
Segment0_Layer:=0
Segment0_Name:=masseter muscle
Segment0_NameAutoGenerated:=0
Segment0_Tags:=Segmentation.Status:inprogress|TerminologyEntry:Segmentation category and type - 3D Slicer General Anatomy list~SCT^85756007^Tissue~SCT^71616004^Muscle~^^^~Anatomic codes - DICOM master list~^^^^^|
Segment1_Color:=0.752941 0.54902 0.513725
Segment1_ColorAutoGenerated:=0
Segment1_Extent:=180 311 184 229 0 0
```

Figure III.7 An example of an NRRD-format mask for a patient, containing the corresponding segmentation information.

Each image in the dataset was thus paired with a corresponding multiclass mask containing detailed localization information for the two muscles. While such segmentation is ideally performed by a medical specialist for maximal precision, we undertook this process ourselves while minimizing potential errors, taking advantage of the fact that both the masseter and medial pterygoid muscles are clearly distinguishable in the cervical region.

At this stage, a corresponding image-mask pair has been successfully created for each image. To ensure the correct alignment between each image and its respective mask, we implemented a Python script for visual overlay **(Appendix 2)**. The resulting superpositions confirmed the accuracy of the annotations **(Figure III.8).**

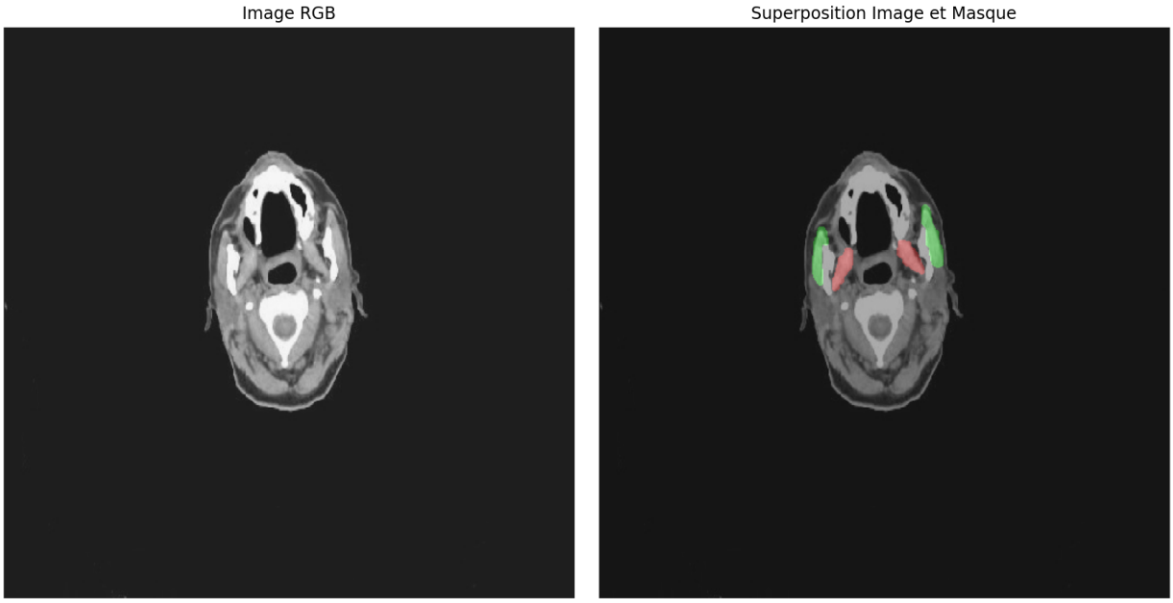


Figure III.8 An overlay result of the segmentation mask on the pre-radiotherapy image for patient HNSCC-01-0004.

Our dataset is now well established, consisting of 50 patients, each with two images, one before and one after treatment. Each image is paired with a corresponding mask that highlights the location of the targeted muscles. In total, we have 100 labelled images.

However, this number is relatively small for training a deep learning model like U-Net, especially in the context of automatic segmentation. This limitation prompted us to adopt an approach to artificially expand our dataset, a process known as ***data augmentation***.

III.2.3 Data augmentation

❖ Definition

To enhance the performance and generalizability of the segmentation model, a set of **data augmentation techniques** which are different transformations (rotation, translation, gaussian noise, brightness adjustment, contrast adjustment, zoom….) was implemented. Using a Python script **(Appendix 3)**, these transformations were applied simultaneously to both the **original medical images** and their corresponding **segmentation masks** in order to synthetically increase the dataset size and reduce overfitting.

The **segmentation mask** is a **label map** that assigns a discrete class (e.g., background, muscle) to each pixel. Therefore, any transformation applied to the mask must **preserve the label values without interpolation artifacts**.

❖ Transformations Applied

A variety of geometric and intensity-based augmentations were performed **(Table III.2)**.

Transformation	Applied to Image	Applied to Mask	Interpolation for Mask	Purpose
Rotation (90°, 180°, 270°)	np.rot90(image, k)	np.rot90(mask, k)	None (discrete rotation)	Introduces rotational variance
Contrast Adjustment	cv2.convertScaleAbs(image, alpha, beta)	Not applied	N/A	Simulates different imaging conditions
Brightness Adjustment	cv2.convertScaleAbs(image, beta=±X)	Not applied	N/A	Simulates illumination variation
Zoom (Scaling)	cv2.resize(..., INTER_LINEAR)	cv2.resize(..., INTER_NEAREST)	INTER_NEAREST	Focuses on different regions
Translation	cv2.warpAffine(..., borderValue=0)	cv2.warpAffine(..., INTER_NEAREST, border=0)	INTER_NEAREST	Mimics anatomical displacement
Gaussian Noise	image + np.random.normal(...)	Not applied	N/A	Improves robustness to noise

Table (III.2) Detailed description of how each transformation was applied to the image-mask pairs.

❖ INTER_NEAREST for Masks

Unlike continuous grayscale images, segmentation masks contain **categorical values** (e.g., 0 for background, 1 for muscle). When resizing or transforming masks, it is critical to use **nearest-neighbor interpolation (INTER_NEAREST)** to preserve these exact label values. Using bilinear or bicubic interpolation would create intermediate values that are medically meaningless (e.g., a pixel value of 0.5 between class 0 and 1), and would lead to incorrect training data for the model.
For example, a mask with values:

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{matrix}$$

If interpolated with linear methods, could become:

$$\begin{matrix} 0.0 & 0.5 & 1.0 \\ 0.0 & 1.0 & 0.7 \\ 0.0 & 0.0 & 0.0 \end{matrix}$$

Whereas INTER_NEAREST ensures label integrity:

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{matrix}$$

This ensures that the neural network learns from accurate and biologically valid segmentation data.

❖ **Overfitting and the Role of Augmentation**

Overfitting occurs when a model performs well on the training data but fails to generalize to unseen data. This usually happens when the model memorizes the training samples rather than learning robust features.

Data augmentation helps to **combat overfitting** by:

- Exposing the model to a wider range of image appearances,
- Forcing it to learn **invariant features** (e.g., recognizing a muscle regardless of its orientation or brightness),
- Simulating variability that may be encountered in real clinical scenarios. [48]

By applying this augmentation strategy, we increased our dataset from 100 to 800 images under various conditions. Each original image generated 7 additional variants, providing a solid dataset size for initial model training. To validate the alignment between the augmented images and their masks, we performed a second overlay for visual inspection, **(Figure III.9)**.

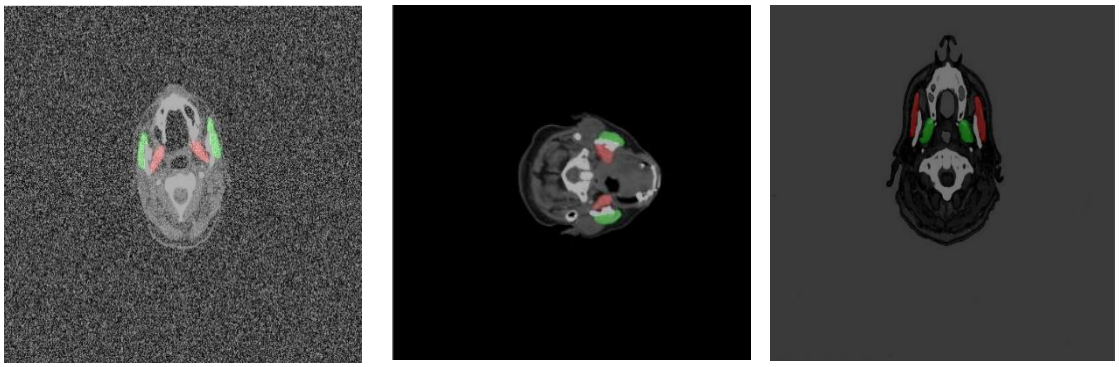


Figure III.9 Some sample images after dataset augmentation with their corresponding masks overlaid.

All these images were grouped within the same folder for subsequent use.

III.2.4 Model Architecture

❖ U-net model

U-Net is a type of convolutional neural network (CNN) specifically designed for image segmentation, that is assigning a class label to each pixel in an image (for example: masseter muscle, medial pterygoid muscle, background...).

Imagine feeding a neck CT image into the network. U-Net's role is to automatically outline the muscles of interest, just like a physician would manually annotate them.

U-Networks like a **symmetric autoencoder**. It consists of two main parts the encoder and decoder, between them are **skip connections** that intelligently link the encoder and decoder layers **(Diagram III.1)**.

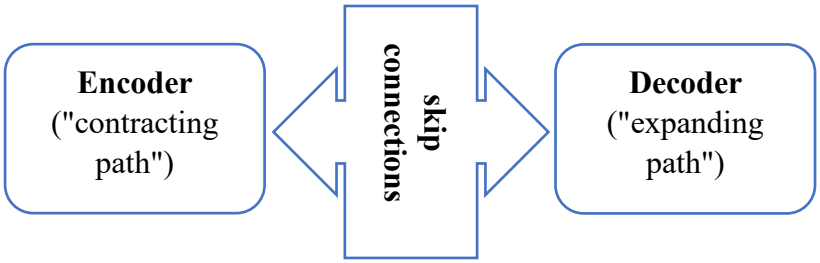


Diagram III.1 A simplified schematic representation illustrating the interaction between the key components of a U-Net architecture, including the encoder, decoder, and skip connections.

❖ Justification for Choosing U-Net

Several reasons support the selection of the U-Net architecture for this work:

- **Proven performance in medical segmentation tasks**, especially for soft tissues, which are often challenging to distinguish in CT images.
- **High spatial accuracy**, enabled by its symmetric encoder-decoder structure with **skip connections**, which help preserve fine details along muscle boundaries.
- **Effectiveness with limited annotated data**, a major advantage in the medical domain where labelled datasets are expensive and time-consuming to obtain.
- **Ease of implementation** using deep learning libraries such as **TensorFlow/Keras**, making it straightforward to integrate U-Net into a full pipeline, from preprocessing to user interface.

Additionally, the predicted segmentation mask can be easily **overlaid on the original image** for clear visualization and clinical interpretation.

❖ Blocks of U-net

➢ Encoder – Understanding the Image

The encoder's job is to analyse the image and detect meaningful patterns. It applies **convolutional filters** to extract features (like edges, shapes…). Then, it uses **max pooling** to reduce the image size and keep only the most important information. With each step, the number of filters increases (so the network learns more complex features), while the spatial resolution decreases.

You go from a high-resolution image (e.g., 512×512) to a compressed, information-rich representation.

- **Convolutional filters**

At the core of each block in the U-Net encoder and decoder is a set of convolutional filters. These filters scan the image by sliding over small patches of pixels and applying a mathematical operation, each pixel value in the patch is multiplied by a corresponding weight in the filter, and the results are summed to produce a single output value.

Mathematically, a convolution is defined as the element-wise multiplication of two matrices, one representing the pixel patch from the input image, and the other representing the filter, followed by the summation of all resulting values **(Figure III.10) (Figure III.11)**.

1	2	3
2	0	0
7	9	1

*

3	2	0
3	0	1
0	5	2

=

1*3 = 3	2*2 = 4	3*0 = 0
2*3 = 6	0*0 = 0	0*1 = 0
7*0 = 0	9*5 = 45	1*2 = 2

Figure III.10 Mathematic convolution [49]

Convolutions over RGB images

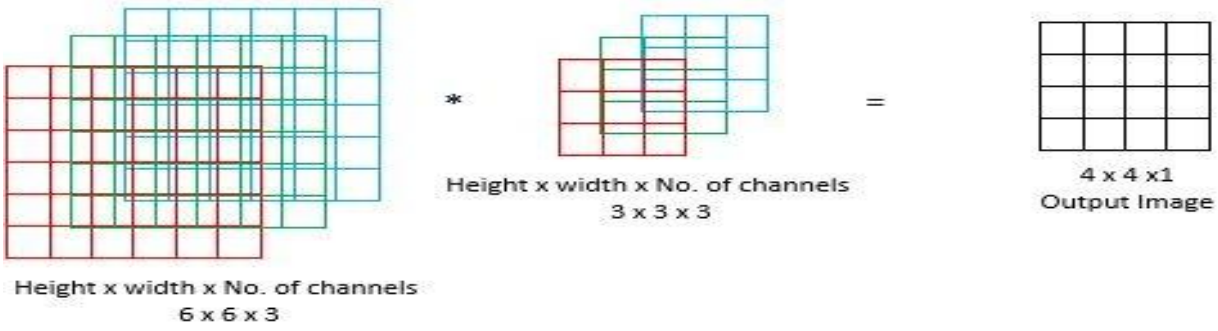


Figure III.11 Convolution over volume [49]

This process is repeated across the entire image to create a new image, called a **feature map**, that highlights specific patterns like edges, textures, or shapes.

- **Filters and Weights**

A **filter** is essentially a small matrix (e.g., 3×3 or 5×5) composed of **trainable weights**. These weights are automatically learned during training: the network adjusts them so that the filter becomes sensitive to meaningful features in the image. Initially, the network has no prior knowledge, the weights are usually random. As it processes more training images and compares its predictions to the ground truth, it gradually updates these weights to improve its ability to detect important patterns. Each element (or pixel) in the filter corresponds to one weight. For example, a 3×3 filter contains 9 elements, which means it has 9 individual weights that are learned and optimized during training. In each convolutional layer, we don't use just one filter, we use **dozens or hundreds** of them in parallel.

- Each filter captures a different type of feature.
- The output of all these filters is stacked together to form a rich representation of the image.
- As we go deeper into the network, filters learn **more abstract features**, at first simple edges, later complex structures like organs or muscles.

Example:

A filter might learn to detect **horizontal edges**, another one might detect **corners**, or **texture patterns** specific to muscles in CT images.

- **Max pooling**

Max Pooling is an operation used in convolutional neural networks (CNNs) to reduce the spatial dimensions of an image while preserving the most important information.

➢ **Example**

A small 4×4 image:

1 3 2 4

5 6 1 2

8 7 3 1

0 9 5 4

If we apply max pooling with a 2×2 window, we divide the image into non-overlapping 2×2 blocks and take the maximum value from each block **(Table III.3)**.

2×2 Block	Max Value
(1, 3, 5, 6)	6
(2, 4, 1, 2)	4
(8, 7, 0, 9)	9
(3, 1, 5, 4)	5

Table III.3 Max pooling method

The resulting image becomes:

6 4

9 5

We've reduced the size from **4×4 to 2×2**, keeping only the strongest values.

Max pooling **reduces data size**, making computations faster, it **preserves dominant features** like edges or textures, and it **helps prevent overfitting** by simplifying the representation.

➢ **Decoder – Reconstructing the Segmentation**

Now that the network knows what is in the image, it needs to figure out where those things are. It gradually **upsamples** the feature maps using **transposed convolutions** (the opposite of the first convolution). At each level, it **merges information from the encoder** via skip connections.

Thanks to this, the network can recover the fine details lost during downsampling.

Example: If a filter in the encoder detected a clear boundary, the decoder can reuse that to precisely place the muscle edge.

➢ **Skip Connections – Bridging Encoder and Decoder**

These direct links between matching encoder and decoder layers allow the network to:

- Preserve important details (like shapes and contours)
- Better localize structures in the image
- Produce highly accurate segmentation maps (with clean muscle boundaries).

After a convolution operation, the output is a **feature map** with raw, linear values. To help the network learn complex, non-linear relationships, we apply a **non-linear activation function**, and **ReLU** is one of the most widely used for this purpose **(Diagram III.2)**.

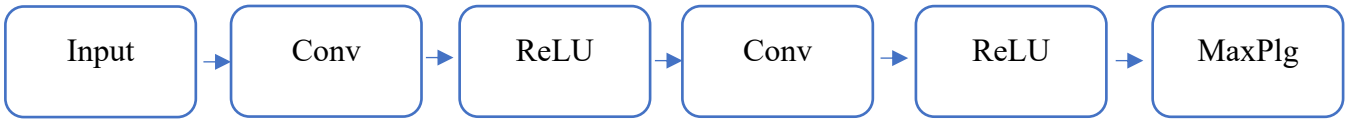


Diagram III.2 Typical flow structure

ReLU stands for **Rectified Linear Unit**, and it's an **activation function** applied after convolution operations to introduce non-linearity into the network.

➢ **Example:**

. If a layer outputs:

-5	1	2
-4	-3	2
1	5	7

After applying ReLU:

0	1	2
0	0	2
1	5	7

ReLU will:

- **Removes negative values**, which often carry no useful signal in images.
- **Speeds up training** and introduces non-linearity
- **Simplifies computations** and avoids unnecessary complexity.

➢ **Output a segmentation map**

At the end, U-Net produces a **prediction map** where each pixel has a probability value (e.g., 0.9 for muscle, 0.1 for background). This mask can be **overlaid on the original image** for visualization.

III.2.5 Training strategy

To implement the U-Net model in our study for the automatic segmentation of both the masseter and medial pterygoid muscles, we designed a specific strategy tailored to this objective. As previously mentioned, the U-Net architecture requires the configuration of several parameters, which will be discussed in the following sections.

❖ **Learning Parameters**

The U-Net model was trained with the following parameters:

- **Input image size:** 256 × 256 pixels, 3 channels (RGB).
- **Number of classes:** 3 (background, masseter muscle, pterygoid muscle).
- **Batch size:** 4 images per training batch.
- **Epochs:** 30 complete passes over the training dataset.
- **Dropout rate:** 0.3, applied to convolutional layers to prevent overfitting by randomly disabling 30% of the neurons during training.
- **Optimizer:** Adam optimizer, an adaptive learning algorithm that adjusts the learning rate during training to speed up convergence and improve performance.

We used also some specific libraries like : **Tensorflow and Keras.**

➢ **Mask Encoding and One-Hot Encoding**

The ground truth segmentation masks are labelled with integers representing the classes:

- 0 for background (encoded as [1, 0, 0])
- 1 for the first muscle class (encoded as [0, 1, 0])
- 2 for the second muscle class (encoded as [0, 0, 1])

This is known as **one-hot encoding** or **categorical encoding**, we use this with «from tensorflow. keras to_categorical », which transforms a single integer label per pixel into a vector of length equal to the number of classes, where only the index corresponding to the class is 1 and the others are 0. This format is required for multi-class classification with softmax output layers and categorical loss functions.

➢ **Dropout Rate Explanation**

Dropout is a regularization technique used to reduce overfitting. During training, dropout randomly sets a fraction (in our model 30%) of input units to zero at each update. This forces the network to learn more robust and redundant features because it cannot rely on any single neuron. At inference time, dropout is disabled, and all neurons are used with scaled weights.

➢ **Optimizer Explanation**

The Adam optimizer combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp. It computes adaptive learning rates for each parameter, based on estimates of first and second moments of the gradients. This allows faster and more stable convergence compared to standard gradient descent.

❖ **Data Flow and Learning Process**

In our U-Net architecture, the model takes four input images. However, since both follow the same processing steps, we'll describe the workflow for one image only.

➢ **Encoder – Downsampling Path**

▪ **Step 1: Convolution Level 1**

- **Input**: image of shape (256, 256, 1) (or (256, 256, 3) for RGB).
- We apply **3 convolutional filters**, each of size 3×3:
  - The filters slide over the image using a 3×3 window.
  - At each window position, we compute a **convolution sum**:

$$\text{value} = \sum_{i=1}^{3}\sum_{j=1}^{3} \text{weight}_{ij} \cdot \text{pixel}_{ij}$$

- This results in **3 feature maps**, one for each filter → shape: **(256, 256, 3)**.
- Apply **ReLU** activation (replaces negative values with zero).
- Apply **Max Pooling 2×2**, reducing size to **(128, 128, 3)**.

➤ **Skip connexion 1** is saved: (128, 128, 3)

▪ **Step 2: Convolution Level 2**

- **Input**: the 3 feature maps from previous level, treated as a 3D volume.
- Apply **5 filters** (3D convolution – filters "see" all 3 channels).
- Output: **5 new feature maps** → shape: **(128, 128, 5)**.
- Apply ReLU + Max Pooling → output size: **(64, 64, 5)**.

➤ **Skip connexion 2** is saved: (64, 64, 5)

▪ **Steps 3 & 4: Continue with same pattern**

- Increase the number of filters at each level (e.g., 128 → 256 → 512).
- Each level halves the spatial dimensions:

$$(64,64,128) \rightarrow (32,32,256) \rightarrow (16,16,512)$$

Save **skip connections** from each level (skip3, skip4).

➢ **Bottleneck (Bridge)**

- Input: output of level 4 → shape: **(16, 16, 512)**
- Apply convolution block with **1024 filters** → output: **(16, 16, 1024)**
- **No further downsampling here** – this is the **deepest part of the network**.

➤ The network now captures **global contextual features**.

➢ **Decoder – Upsampling Path**

The decoder mirrors the encoder to **reconstruct the segmented image**:

- **Step 1: Upsampling + Fusion**
- Upsample the bottleneck output:

$$(16,16,1024) \rightarrow (32,32,512)$$

- **Concatenate** with skip4 (from encoder level 4).
  - This merges **fine details** with **deep features**.
- Apply **two convolutional layers** to refine and reconstruct the structure.
- **Subsequent Decoder Levels**
- Continue the upsampling and concatenation process:

$$(32,32,512) \rightarrow (64,64,256) \rightarrow (128,128,128) \rightarrow (256,256,64)$$

- At each step, concatenate with the corresponding skip connection (skip3, skip2, skip1).

➢ **Final Output Layer**

- A **final 1×1 convolution** reduces the number of channels to **3** (one for each class: background, muscle 1, muscle 2).
- Apply **Softmax activation** on each pixel:
  - Each pixel becomes a probability vector: [p0,p1,p2].These values satisfy:

$$\sum_{i=0}^{2} p_i = 1$$

  - The predicted class is:

predicted_class=argmax(pi)

Consider a 6x6 grayscale image as an example:

$$\text{Image} = \begin{bmatrix} 1 & 2 & 0 & 3 & 1 & 2 \\ 4 & 1 & 1 & 0 & 2 & 3 \\ 1 & 3 & 2 & 1 & 1 & 0 \\ 0 & 2 & 3 & 2 & 4 & 1 \\ 1 & 1 & 0 & 2 & 3 & 1 \\ 3 & 0 & 2 & 4 & 1 & 2 \end{bmatrix}$$

By applying a 3x3 kernel

$$\text{Filter} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The convolution result is calculated as follows:
$(1\times0) + (2\times1) + (0\times0) + (4\times1) + (1\times(-4)) + (1\times1) + (1\times0) + (3\times1) + (2\times0) = 2 + 4 - 4 + 1 + 3 = 6.$

This value, 6, becomes the first element of the feature map.
We then slide the filter one pixel to the right and repeat the operation across the entire image.

In this example, the resulting feature map has a resolution of 4×4.

To maintain the original resolution after convolution, we use "padding = same", which adds rows and columns of zeros around the image. This ensures that the output feature maps remain the same size as the input.

Let's now assume the resulting feature map looks like this:

$$\text{Map1} = \begin{bmatrix} 6312 \\ 0521 \\ 4302 \\ 1213 \end{bmatrix}$$

If we apply 3 different filters, we obtain 3 feature maps, resulting in a feature map of size

$$(4, 4, 3)$$

Next, we apply the ReLU activation function to replace all negative values with zero.

Then, in the first block, we perform max pooling using a 2×2 window to reduce the spatial dimensions.

$$\begin{bmatrix} 6 & 3 \\ 0 & 5 \end{bmatrix} = 6$$

The resulting feature map after max pooling will be:

$$\begin{bmatrix} 6 & 2 \\ 4 & 3 \end{bmatrix}$$

This operation is applied to all feature maps, resulting in a tensor of size (2, 2, 3) for the corresponding skip connection from the first layer.

We repeat this process for each layer of the encoder.

In the decoder phase:

- We first apply upsampling, e.g., from (2, 2, 5) to (4, 4, 5).
- Then we concatenate the upsampled map with the corresponding skip connection. If the skip connection is of shape (4, 4, 3), the concatenated tensor becomes:

$$(4, 4, 5) \oplus (4, 4, 3) = (4, 4, 8)$$

Next, we apply convolutional layers to reduce this back to the desired number of classes.

Final Output: (6, 6, 3)

Each pixel in the output represents a probability distribution over the 3 classes. For example,

Pixel(i, j) = [0.8, 0.1, 0.1] ⇒ class = 0 (the class with the highest probability)

The dataset is split into two parts: 90% for training and 10% for validation, corresponding to 720 images for training and 80 for validation.

This process is repeated for 30 epochs. At the end of each epoch, the loss function is computed to evaluate the model's performance and to monitor for potential overfitting.

➢ **Loss Function**

To effectively train the U-Net model for multi-class segmentation, a combined loss function was employed:

- **Categorical Cross-Entropy Loss:** Measures the pixel-wise difference between predicted class probabilities and ground truth masks.

  True label for a pixel = [0 1 0] (class 1)

  Predicted = [0.2 0.7 0.1]

  CCE loss = -log(0.7) ≈ 0.357

- **Dice Loss:** Derived from the Dice coefficient, it directly maximizes the overlap between predicted and true segmentations, which is particularly useful for imbalanced classes or small objects.

  Dice Coefficient formula:

$$\text{Dice}(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

  In segmentation:

$$\text{Dice} = \frac{2\sum(y_{\text{true}} \cdot y_{\text{pred}})}{\sum y_{\text{true}} + \sum y_{\text{pred}}}$$

- Dice = **1**: perfect match → Loss = 0
- Dice = **0**: no overlap → Loss = 1

The **combined loss** is defined as the sum of the categorical cross-entropy and Dice loss, enabling the model to learn more accurate and robust segmentation boundaries.

❖ **Visualization of Learning Curves**

During training, the evolution of the loss and Dice coefficient on both training and validation sets was monitored and plotted over epochs. These learning curves help to:

- Verify the model is converging properly.
- Detect overfitting or underfitting by comparing training and validation metrics.
- Adjust hyperparameters if necessary.

Typically, the loss decreases while the Dice coefficient increases, indicating improvement in segmentation accuracy.

III.2.6 Post-processing

To visually assess the performance of our U-Net model, we implemented a visualization routine that displays segmentation results for a selected test image. The trained model is used to predict the corresponding mask, and both the ground truth and the prediction are processed for comparison.

The **ground truth mask**, originally in a one-hot encoded format, is converted into a matrix of class indices using the argmax function. This operation reduces each pixel's one-hot vector into a single integer representing the class label.

Similarly, the **predicted mask**, which is also a set of probability vectors over all classes, is transformed using argmax to select the most likely class for each pixel.

A **custom colormap** is applied for visualization:

- **Black** represents the background (class 0),
- **Green** represents the masseter muscle (class 1),
- **Red** represents the medial pterygoid muscle (class 2).

The output is displayed in three side-by-side panels:

1. The original input image,
2. The ground truth mask,
3. The predicted mask.

This layout enables a direct and intuitive comparison of the segmentation quality.

In semantic segmentation tasks, one-hot encoding is commonly used to represent class labels at the pixel level.

Once the segmentation results are validated, we can proceed to the second objective, which is surface area calculation.

III.2.5 Surface Area Computation and Nutritional Recommendation System

The detection of muscle atrophy relies on the **quantitative evaluation** of the surface areas of the **masseter** and **medial pterygoid** muscles before and after radiotherapy (RT). Once the segmentation masks are generated by the U-Net model, a simple yet effective method is used to estimate the area occupied by each muscle in the CT scan images.

❖ **Methodology**

The segmentation masks are grayscale images where each pixel is assigned to a class:

- **0**: Background
- **1**: **Masseter** muscle
- **2**: **Medial pterygoid** muscle

To compute the surface area of a specific muscle, the algorithm counts the number of pixels belonging to the corresponding class in the mask:

$$\text{Muscle Area} = \sum_{i,j} (\text{mask}_{i,j} == \text{class index})$$

This operation is implemented via the function calculate_surface(mask, class_index) in the script. It returns an integer representing the muscle surface in **pixels**.

❖ **Compute masseter and pterygoid areas separately**

This is handled via **semantic segmentation** using class labels in the mask predicted by the U-Net model.

This line does the key work:

mask = np.argmax(prediction, axis=-1)

- The U-Net model outputs a prediction tensor of shape (256, 256, 3) (3 channels for 3 classes):
- np.argmax(prediction, axis=-1) creates a 2D mask where each pixel has the class label: 0, 1, or 2.

Then it computes areas like this:

def calculate_surface(mask, class_id):

return np.sum(mask == class_id)

- mask == 1 → counts pixels for the **masseter**
- mask == 2 → counts pixels for the **pterygoid**

So calculate_surface (mask, 1) gives the number of **masseter pixels**, and calculate_surface(mask, 2) gives the **pterygoid pixels**.

❖ **Before/After RT Comparison**

Each patient has two images: one acquired **before** and one **after** radiotherapy. The same segmentation process is applied to both. The muscle areas are computed for both timepoints and compared:

- A **decrease in surface area** is considered an **indicator of muscle atrophy**.
- The presence of atrophy is confirmed if **at least one of the two muscles** shows a reduction.

This binary evaluation is used to trigger the **personalized nutrition recommendation system**.

❖ **Nutritional Decision Support**

To compensate for possible muscle loss, a **custom dietary recommendation** is generated based on both:

The **presence of atrophy**, and the patient's **BMI** (Body Mass Index):

$$\text{BMI} = \frac{\text{Weight (kg)}}{(\text{Height (m)})^2}$$

Depending on the atrophy status and BMI category, a **recommendation is proposed**, following clinical nutrition guidelines **(Table III.4)**.

Condition	Dietary Recommendation
Atrophy + BMI < 18.5	High-protein, high-calorie diet (2200–2500 kcal/day) with increased antioxidant intake (vitamins C, E, selenium)
Atrophy + 18.5 ≤ BMI < 25	2200 kcal/day, moderate exercise, protein-rich meals, and antioxidant support (e.g., berries, leafy greens)
Atrophy + BMI ≥ 25	Calorie-controlled diet (~2000 kcal), lean proteins, and foods rich in antioxidants to combat inflammation
No atrophy + BMI < 18.5	Moderate increase in protein and calories, plus natural antioxidant sources (e.g., citrus fruits, nuts)
No atrophy + 18.5 ≤ BMI < 25	Maintenance diet (2000 kcal/day, 1.2 g protein/kg), include antioxidant-rich foods for cellular protection
No atrophy + BMI ≥ 25	Light, high-fiber diet (1800 kcal/day) with antioxidant support to promote healthy metabolism and reduce stress

Table III.4 Nutritional Recommendations Based on Muscle Atrophy and BMI

As previously discussed, radiotherapy increases oxidative stress in the body, which can lead to inflammation and subsequent muscle atrophy.

By combining a protein-enriched diet with antioxidant-rich foods, we help ensure that muscle proteins are properly synthesized and protected, thus improving recovery and reducing tissue damage.

This rule-based engine integrates both the image analysis and the physiological state of the patient to propose a personalized nutritional strategy.

III.2.7 Interface Development

To provide a complete and intuitive platform, a user-friendly interface was developed using **React.js**, **Tailwind CSS**, **ShadCN UI**, and **Framer Motion**. This responsive web interface allows seamless interaction for both technicians and doctors involved in the patient care workflow.

❖ **Global Structure and Workflow**

The interface is structured as follows:

- **Home Page**: Animated landing page featuring the project branding and medical theme visuals.
- **Role Selection**: Users choose their role (Doctor or Technician) to access personalized dashboards.
- **Technician Dashboard** :
    - Upload and manage patient information and CT scans.
    - Launch the U-Net segmentation model.
    - View segmentation results and calculate surface changes.
    - Generate personalized nutrition plans.
    - **New features**:
        - **Live doctor availability**: Technicians can see which doctors are online for real-time collaboration.
        - **Emergency notification**: Quick alert button in case of critical patient data or results.
        - **Model statistics**: Display of segmentation model performance (accuracy, dice score, etc.).
- **Doctor Dashboard** :
    - View and validate patient segmentation reports and plans.
    - Download and print reports.
    - **New feature**: **Integrated chat** with technicians for instant communication.
- **Additional sections** :
    - **About**: Overview of the project objectives, medical context, and technological approach.

❖ **Workflow Summary**

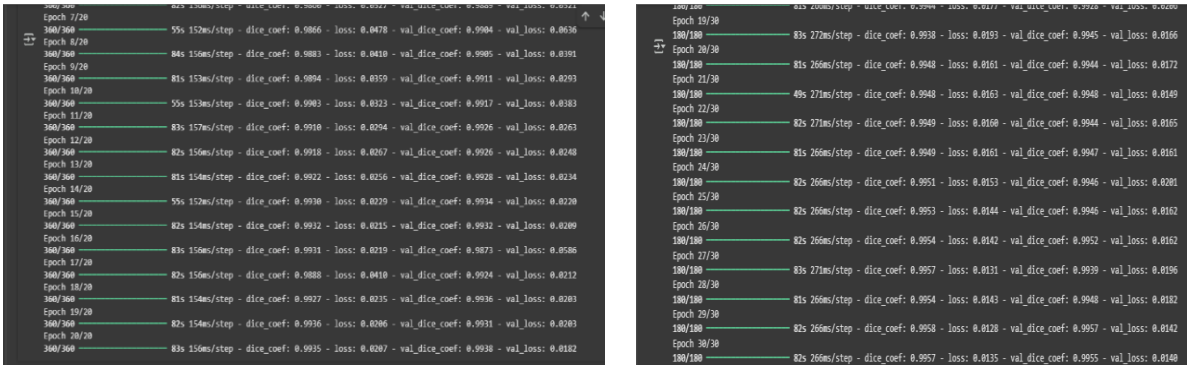
1. Technician uploads patient data and medical images.
2. Segmentation and muscle analysis are performed.
3. Results are used to generate a nutritional plan.
4. Doctors review the plan, provide feedback, or validate it.
5. Communication and collaboration are streamlined via integrated chat and status indicators.

III.3 Results

III.3.1 Segmentation Performance

To achieve our objective, we applied the previously described segmentation process to our patient data. The first step was to evaluate the model's ability to accurately segment the target muscles, by analyzing whether it performs precise delineation and whether the evaluation metrics indicate proper learning without overfitting or underfitting. This step was essential in determining whether muscular atrophy was present or not.

During training, we closely monitored the model's performance metrics. We observed that with each epoch, the Dice coefficient steadily increased while the Dice loss and categorical cross-entropy loss decreased for both training and validation datasets. This trend is a strong indicator of good model learning and generalization **(Figure III.12)**.



(A) (B)

Figure III.12 Resulting Loss and Dice coefficient during training model (A) two images simultaneously for 20 epochs, (B) four images simultaneously for 30 epochs

We tested two training approaches. In the first **(A)**, the model was trained on two images simultaneously for 20 epochs. The best result achieved with this configuration was a Dice coefficient of **0.9935** and a validation loss of **0.0207**.

In the second **(B)** we improved performance, we modified the approach by training the model on four images at a time for 30 epochs. This adjustment resulted in a Dice coefficient of **0.9957** and a validation loss of **0.0135**, indicating better training and generalization.

Based on these results, we selected and validated the second model configuration (4 images, 30 epochs) for the remainder of the study.

We also plotted these metrics as a function of epochs to identify the point at which the model achieved optimal performance **(Figure III.13)**.

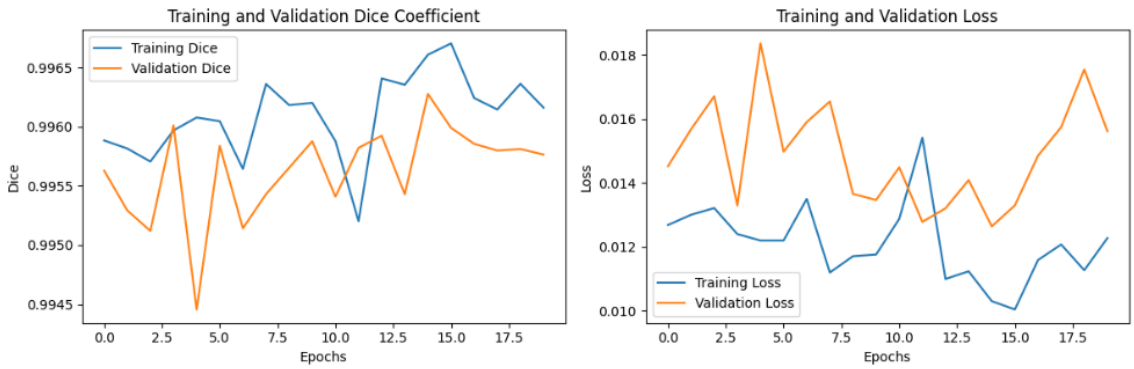


Figure III.13 Training and validation Loss and Dice Coefficient Curve

The graphs above illustrate the evolution of the **Dice Coefficient** and **Loss** during training and validation over 20 epochs.

- **Dice Coefficient (Left Plot)**

The **Dice coefficient for the training data** (blue curve) shows a consistently high performance, fluctuating slightly but stabilizing around **0.996**, which indicates a very accurate segmentation. The **validation Dice coefficient** (orange curve) closely follows the training curve, with values consistently above **0.995**, suggesting that the model generalizes well to unseen data. The similarity between the two curves demonstrates **no significant overfitting**, which confirms that the model maintains good performance beyond the training set.

- **Loss (Right Plot)**

The **training loss** (blue curve) decreases progressively and reaches values around **0.011**, indicating improved learning across epochs. The **validation loss** (orange curve) remains slightly higher (approximately **0.014 to 0.018**) but within an acceptable range. The relatively small gap between the training and validation losses supports the conclusion that the model is **neither overfitting nor underfitting**.

These results highlight the **robustness and reliability of the model**. With a Dice coefficient greater than 0.995 and consistently low loss values, the model demonstrates excellent segmentation performance and strong generalization capacity. Therefore, this configuration (4 images processed simultaneously over 30 epochs) was **validated as optimal** for further analysis and practical application in muscle segmentation tasks.

Finally, we compared the predicted segmentation masks with the ground truth masks to visually validate the model's accuracy and reliability **(FigureIII.14) (Figure III.15)**.

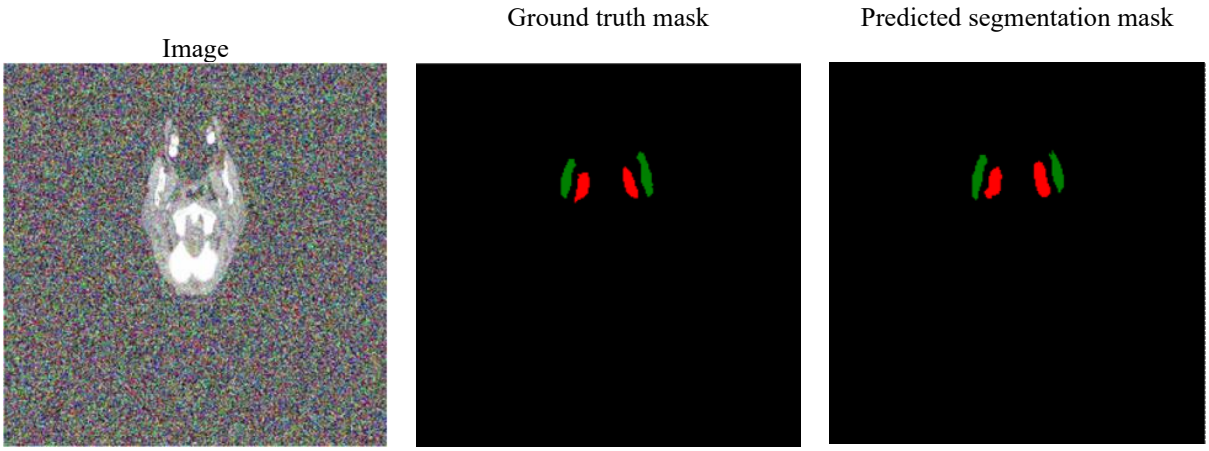


Figure III.14 Prediction of the mask for the third validation image using the trained model.

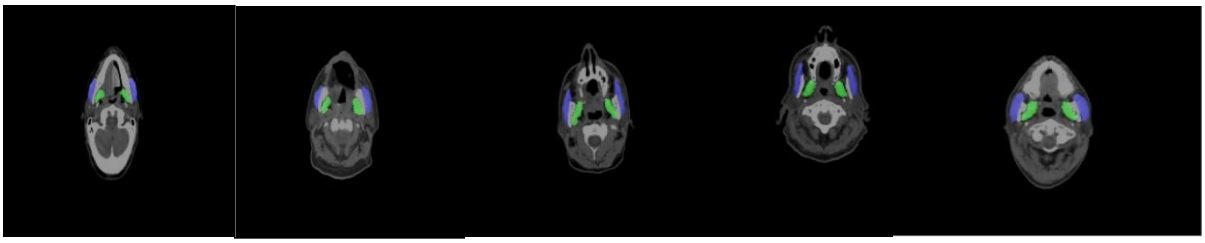


Figure III.15 Applying the trained model to segment randomly selected images.

In **(Figure III.14)**, we compare the ground truth masks with the predicted masks for a noisy test image. Despite the presence of significant visual noise, the predicted segmentation closely resembles the real mask, indicating that the model generalizes well even in challenging scenarios. The structures are accurately identified in both shape and location, with minimal discrepancies between the predicted and actual masks.

To qualitatively assess the performance of our trained U-Net model, we applied it to segment randomly selected CT scan images from both pre-treatment and post-treatment datasets. As shown in **(Figure III.15)**, the model was able to correctly localize and segment the masseter and medial pterygoid muscles (in blue and green, respectively) across multiple samples. The visual overlay demonstrates the model's robustness in detecting anatomical structures under varying conditions and patient cases.

These visual outcomes confirm that the trained model is capable of producing reliable segmentations, which supports the quantitative performance metrics observed previously.

III.3.2 Surface Area Analysis

In the second stage of our study, we focused on **calculating the surface area of the masseter and medial pterygoid muscles** to detect **muscle atrophy in this region**. As previously explained, such atrophy, when caused by radiotherapy, can lead to **trismus**, which limits mouth opening and **impairs nutrition**. Our goal was to **test the feasibility and usefulness of our model** in identifying, alleviating, or potentially preventing these side effects.

After applying the selected methods, we successfully segmented the target muscles. We then calculated the surface area of each muscle for every patient:

- using the **"before" image**, we measured the surface area of the masseter and pterygoid muscles,
- and did the same with the **"after" image**.

Rather than computing physical surface areas in mm², we **counted the number of pixels** classified as each muscle. If the pixel count for a given muscle was lower in the "after" image than in the "before" image, we considered it an indicator of **muscle atrophy (Table III.5)**.

We were then able to determine the **percentage of patients in whom atrophy was detected**, and those in whom it was not. These results were **compared to a semi-automatic method** using **adaptive thresholding**, to evaluate how well our model performed in comparison.

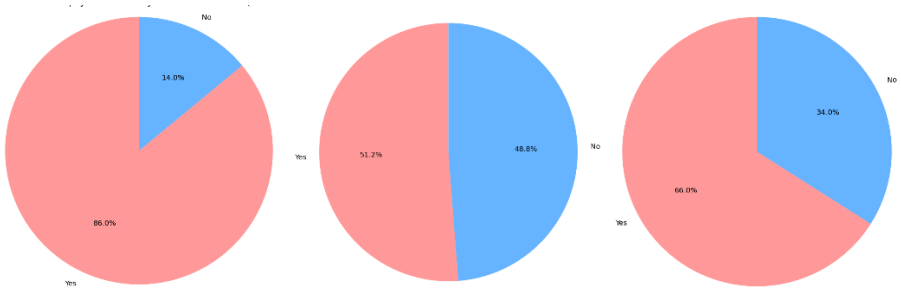
Additionally, based on the Excel data provided by the dataset which included an **estimate of total body muscle mass** we identified cases of general muscle atrophy. Among these, we determined **how many patients specifically showed atrophy in the masseter or pterygoid muscles**, rather than in other muscle groups **(Figure III.19)**.

Patient	SMB	SPB	SMA2	SPA2	AtrophyUNet
HNSCC-01-0004	232	227	241	238	0
HNSCC-01-0005	313	783	364	440	1
HNSCC-01-0006	245	188	367	266	0
HNSCC-01-0014	436	326	263	319	1
HNSCC-01-0016	549	345	473	266	1
HNSCC-01-0019	353	283	275	166	1
HNSCC-01-0027	300	311	297	289	1
HNSCC-01-0029	410	599	346	516	1
HNSCC-01-0082	414	315	397	331	1

Table III.5 Table of patient samples for which muscle surface area in pixels was calculated using U-Net, along with the detection (or absence) of atrophy, **SMB**: Surface area of the masseter muscle before treatment, **SMA**: Surface area of the masseter muscle after treatment, **SPB**: Surface area of the medial pterygoid muscle before treatment, **SPA**: Surface area of the medial pterygoid muscle after treatment.

According to the results shown in **(Table III.5)**, muscular atrophy was detected in some patients, while it was not observed in others. The location of the atrophy also varied: in some cases, it was identified only in the masseter muscle, such as in patient HNSCC-01-0082; in others, it appeared in the medial pterygoid muscle, like in patient HNSCC-01-0005; and in certain cases, atrophy was present in both muscles, as seen in patient HNSCC-01-0016.

These results were used to calculate comparative percentages between the U-Net method and the thresholding method. Based on the muscle mass values extracted from the files, atrophy was predicted when the post-treatment muscle mass was lower than the pre-treatment value **(Figure III.16)**.



(A) (B) (C)

Figure III.16 Comparison of Atrophy Detection Rates: U-Net vs. Adaptive Thresholding and Cross-Validation with Whole-Body Muscle Data, **(A)** Prevalence of muscle Atrophy based on body cross-sectional area **(B)** Prevalence of muscle atrophy based on masseter and medial pterygoid muscles using thresholding method **(C)** Prevalence of muscle atrophy based on masseter and medial pterygoid muscles using U-net model.

As shown in the graphs **Figure(III.16)**, atrophy was detected using the thresholding method in **51.2%** of patients, while **48.8%** showed no detectable atrophy. With U-Net-based segmentation, atrophy was identified in **66%** of patients, and not detected in **34%**. This indicates the superiority of the automated segmentation method using U-Net. It is important to note that this comparison was performed on the same patient population.

Assuming that these results are reliable and that the muscle mass values obtained from the Excel file are accurate, we observed that **86%** of patients presented muscular atrophy, while only 14% did not. Since this overall muscle mass includes the entire body, we can infer that among the 86% of patients with atrophy, **66%** showed cervical (neck-level) muscle atrophy involving the masseter and pterygoid muscles. The remaining **20%** of patients likely exhibit atrophy in other regions of the body.

- **Conclusion:**

These results demonstrate that cervical muscle atrophy is present in a significant majority of the analyzed patients, with an overall prevalence of 86% muscle atrophy in the studied population. Among these patients, 66% exhibited localized atrophy in the neck muscles (masseter and pterygoid), detected with higher accuracy using the automated U-Net segmentation method, compared to the thresholding method which identified atrophy in only 51.2% of the patients.

This difference highlights the superior sensitivity and reliability of U-Net segmentation for assessing cervical muscle atrophy. Furthermore, the strong correlation between local muscle mass reduction and overall muscle mass loss suggests that segmented analysis of cervical muscles can serve as a relevant indicator of general muscle degradation in post-radiotherapy patients.

Thus, the U-Net method proves to be a promising tool for clinical monitoring of muscle atrophy in this context, with a localized atrophy detection rate reaching **66%**, enabling better targeted nutritional and therapeutic management.

III.3.3 Interface result

The user begins on the NutrAdio home page **(Figure III.17)**, which includes a brief introduction and a ***'Start'*** button to proceed to the next step. A second button, labeled ***'About'***, provides a general overview of the project **(Figure III.18)**.

Figure III.17 Home page

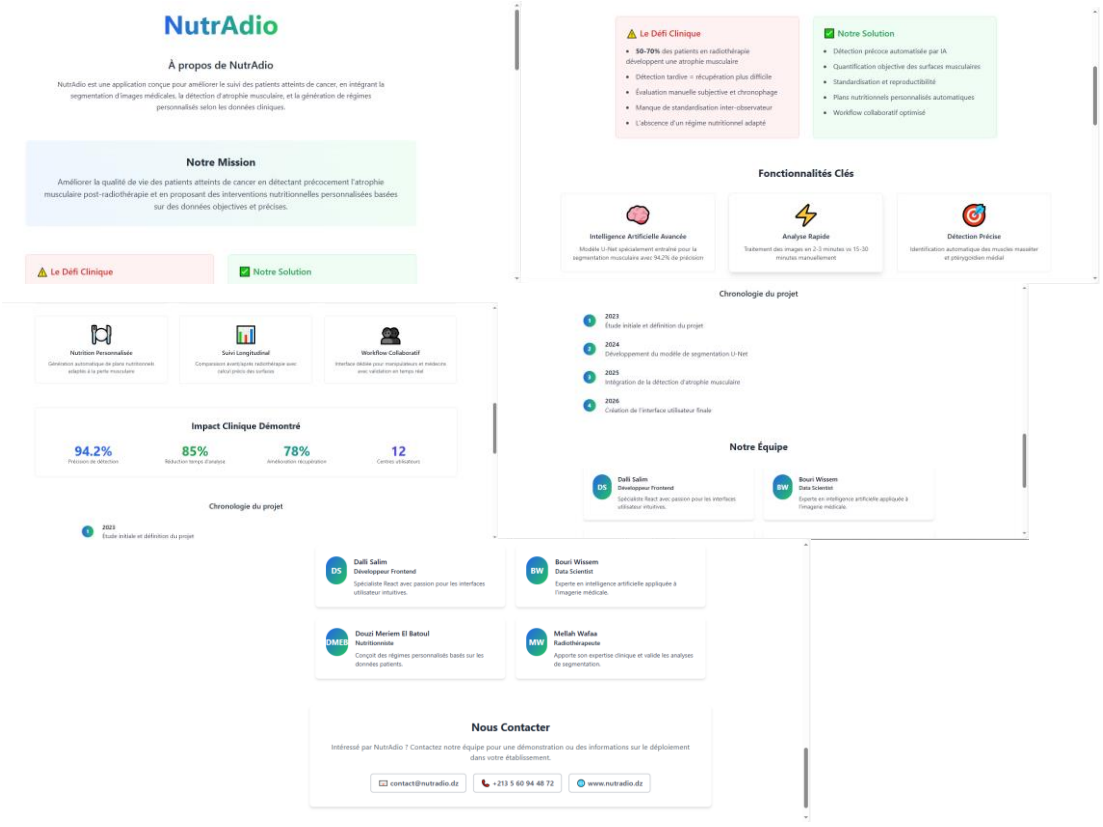


Figure III.18 Apropos page

Next, the user proceeds to the profile selection screen, choosing between ***'Doctor'*** or ***'Technician'***, followed by a login interface **(Figure III.19)**.

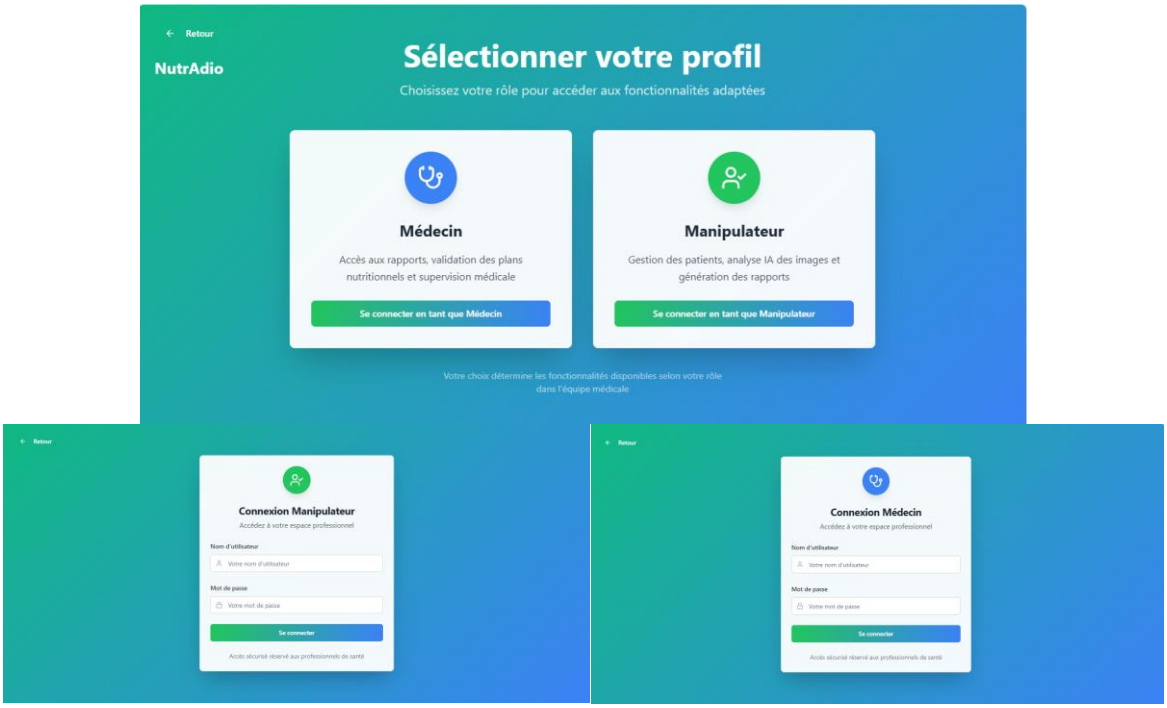


Figure III.19 Profil selection and login

1) Technician Dashboard

In the technician's dashboard, the user begins by entering the patient's information (name, surname, weight, etc.) to create a new patient record **(Figure III.20)**. Then, the user upload the patient's images befor radiotherapy and after radiotherapy and he launches the AI analysis, which performs automatic muscle segmentation **(Figure III.21)**.

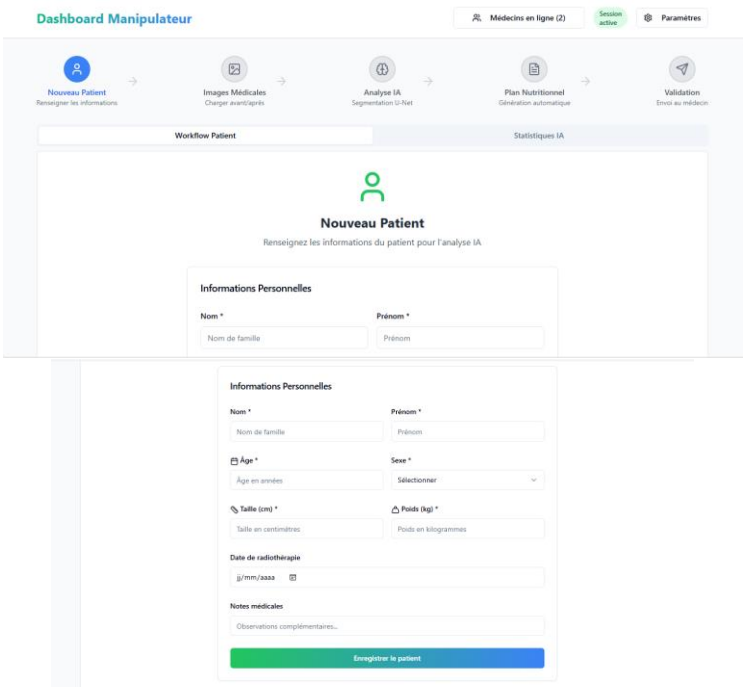


Figure III.20 Patient registration

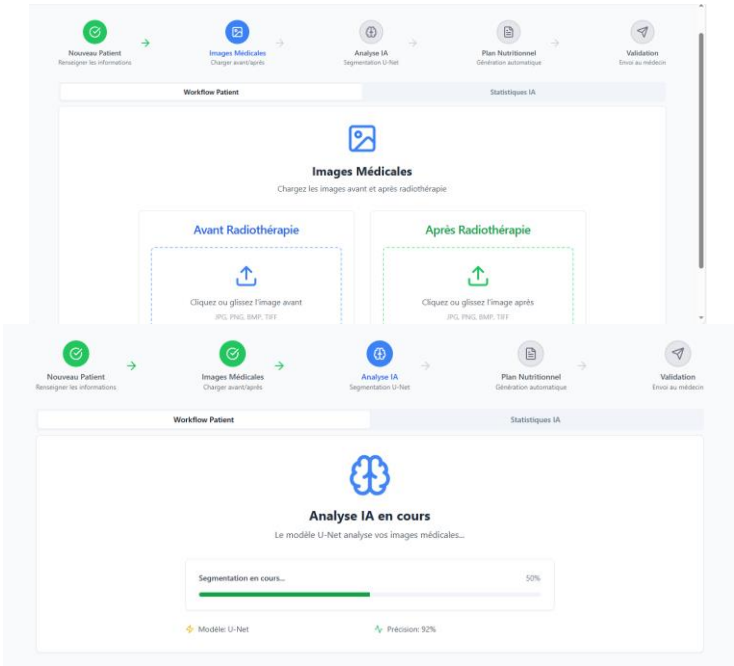


Figure III.21 AI analysis

After a few seconds, the results are displayed, showing the muscle surface areas before and after radiotherapy for the masseter and medial pterygoid muscles, along with an identification of the presence or absence of atrophy **(Figure III.22) (Figure III.23)**. Based on this diagnosis and the initial data such as weight and height, the BMI is calculated. From these, appropriate nutritional recommendations are generated. Finally, a complete report is sent to the doctor for validation **(Figure III.24)**.

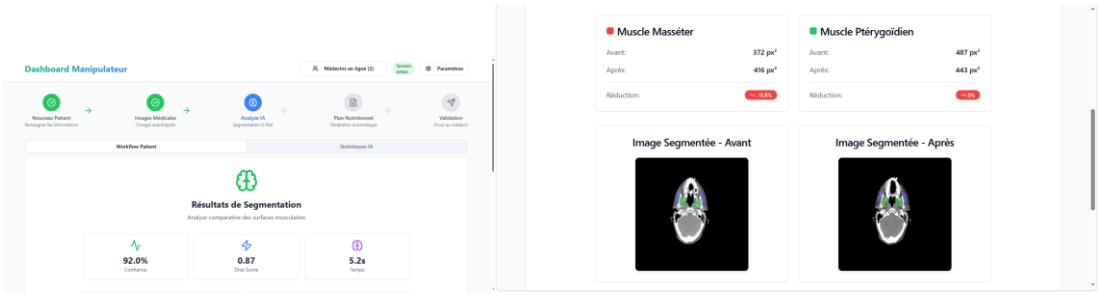


Figure III.22 Segmentation results

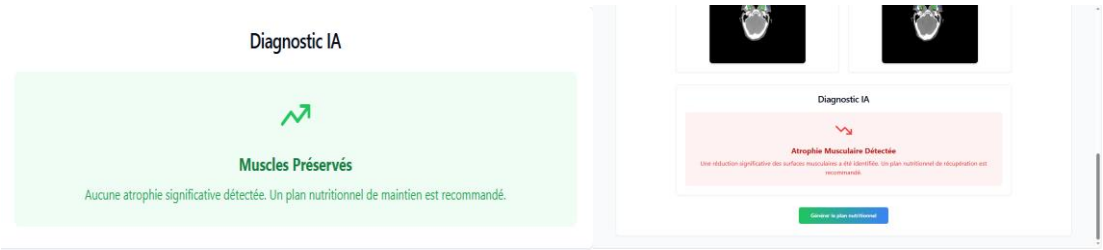


Figure III.23 Atrophy detection

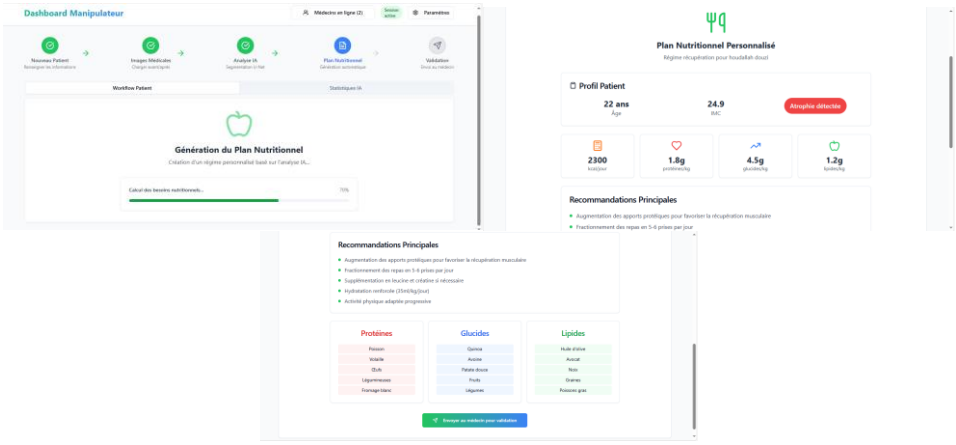


Figure III.24 Personalized dietary recommendation

There is also an option to send an emergency alert to an available doctor.

2) Doctor dashboard

In the doctor's dashboard, the physician receives reports sent by the technicians **(Figure III.25)**. They can review, validate, or modify the reports, as well as download them. The dashboard provides access to reports awaiting validation, validated cases, patients diagnosed with atrophy, and an archive of past patients **(Figure III.26)**.

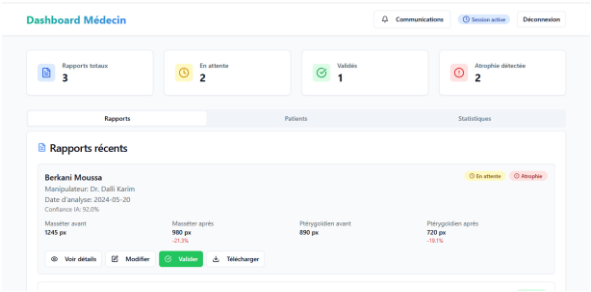


Figure III.25 Doctor Dashboard

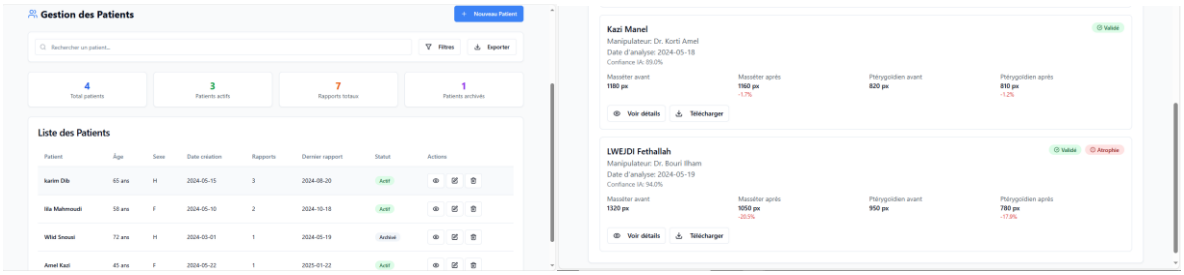


Figure III.26 Patients management

It also includes a statistics section and an AI evaluation module **(Figure III.27)**. Additionally, the doctor's dashboard features a communication section for discussions between doctors.

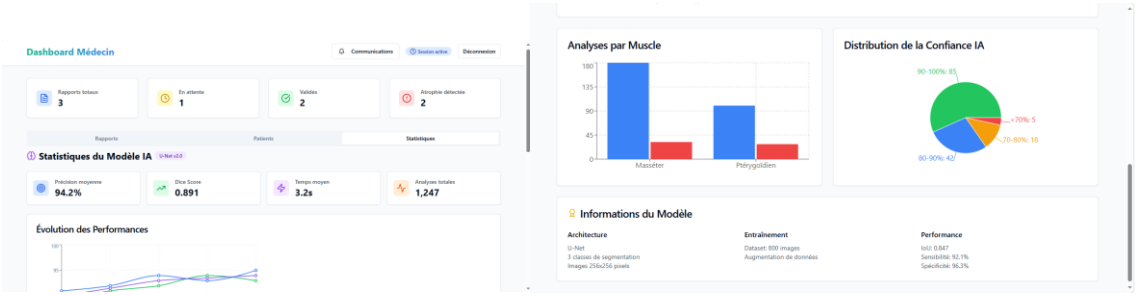


Figure III.27 AI statistics

III.4 Discussion

III.4.1 Critical analysis of the segmentation results

During our initial attempt using adaptive thresholding, the goal was to begin with a simple approach before moving on to more complex methods. We tried to segment the target muscle the masseter and medial pterygoid,using semi-automatic techniques. While the results were not outstanding, this approach still provided some noteworthy outcomes **(Figure III.28)**. For instance, we were able to extract the target muscles from cervical cross-sections that also contained various other structures, such as other muscles, fat, and glands, all with similar contrast and texture characteristics.

However, this method tended to distort the actual shape of the muscles, which led to inaccurate surface area calculations, as the segmentation did not reflect the real anatomical boundaries. Due to this high margin of error, it became clear that we could not build our study on flawed segmentations and unreliable surface measurements.

This limitation prompted us to adopt a more convincing and strategic approach. Given the proven success and growing popularity of CNN-based models, particularly U-Net, we decided to implement this deep learning strategy.

The results obtained with the U-Net model demonstrated a satisfactory ability to automatically segment the masseter and medial pterygoid muscles in cervical medical images. The Dice coefficient and loss scores on the test set confirmed the model's overall strong performance, enabling accurate and rapid detection of the muscles. However, in a few cases, we observed some segmentation issues, such as muscle inversion or over-segmentation of the pterygoid muscle **(Figure III.29)**. However, errors occurred in 2 out of 10 images, representing a 20% rate, which cannot be considered negligible. This highlights the need for further refinement of the model to improve reliability and accuracy, especially in clinical contexts.

Visual inspection of the predicted masks confirmed that the model accurately followed muscle contours in most cases. Nonetheless, some under-segmentation or over-segmentation errors were noticed in edge cases, especially when the muscles were severely atrophied, the image quality was poor, or contrast levels were suboptimal.

These findings suggest that although the model is robust, further refinement, such as enhanced preprocessing, data augmentation, or architecture tuning, could help improve generalization and reduce error cases.

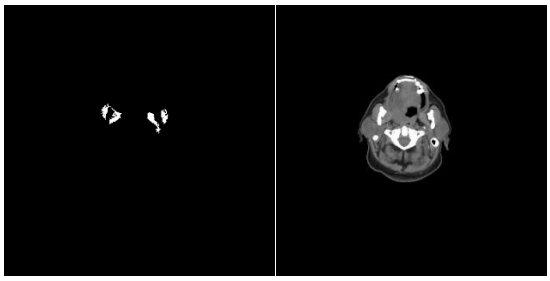
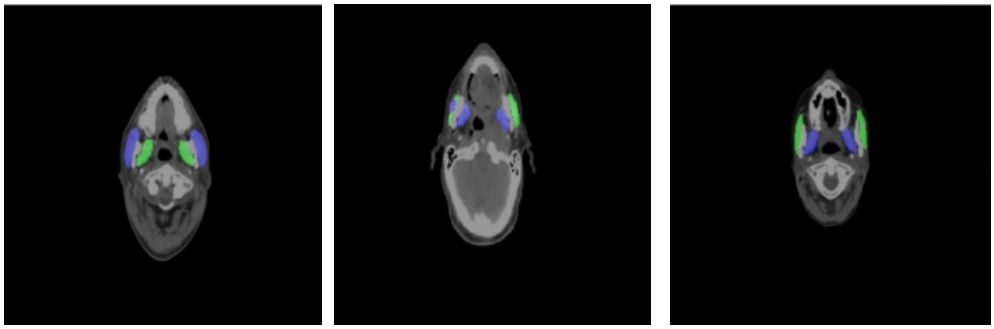


Figure III.28 Masseter and medil pterygoid muscles segmented with thresholding approach



(A) (B) (C)

Figure III.29 Segmentation of the masseter and medial pterygoid muscles using the U-Net model: (A) correct segmentation, (B) over-segmentation of the medial pterygoid with color inversion, (C) color inversion between the two muscles.

III.4.2 Model Limitations

Several limitations were identified during this study:

- **Data Quality and Quantity**:

  The model's performance is highly dependent on the quality of the input images and the accuracy of the initial annotations. Our dataset, being open-source and dating back to the 1990s, included images acquired with older CT scanners, which presented a major challenge in terms of resolution and contrast quality. Moreover, the dataset by us contained only 100 images. Although we expanded the dataset to 800 images through data augmentation techniques, the overall number remains limited and should be increased to improve robustness and generalization.

- **Learning Bias**:

  If most images originate from a single scanner type or medical institution, the model may develop inherent biases and perform less effectively on images acquired in different clinical settings or with different imaging protocols.

- **Generalization Capacity**:

  Applying the model to patients outside the training dataset, especially those with anatomical anomalies or advanced post-treatment changes, requires broader validation. Another significant limitation is that annotations were not performed by a medical expert. Due to resource constraints, the muscle masks were manually annotated without expert supervision, increasing the risk of inaccuracies despite careful efforts to preserve anatomical correctness.

III.4.4 Clinical Relevance of Atrophy Detection and Nutritional Recommendation

The automatic detection of muscle atrophy represents a promising clinical tool, particularly for the follow-up of patients after radiotherapy. It enables early identification of individuals who require targeted nutritional intervention to prevent excessive muscle mass loss. When it comes to the neck region, as previously discussed, muscle atrophy, especially in the masseter and pterygoid muscles, is a major contributing factor to trismus, a condition that significantly impairs both nutrition and quality of life.

Integrating a personalized dietary recommendation system based on segmentation results allows anatomical data to be translated into actionable clinical decisions. This enhances the relevance of the tool within a multidisciplinary care pathway and supports more effective and individualized patient management.

III.4.5 Benefits of the Developed Interface

The interface developed in this project plays a central role by providing fast and intuitive access to segmentation results, surface calculations, and nutritional recommendations. It facilitates seamless collaboration between technicians and physicians through clear, interactive dashboards tailored to each user profile.

Moreover, the automation of the entire workflow,from image import to the generation of personalized dietary plans,significantly reduces workload while ensuring standardization of clinical practices. The modern design, integrated animations, and user-friendly ergonomics further enhance the clinical adoption and usability of the tool.

CONCLUSION

Conclusion

Cancer remains one of the leading causes of morbidity and mortality worldwide. Advances in treatment modalities, such as surgery, radiotherapy, and chemotherapy, have significantly improved patient survival. However, these treatments are not without side effects. Among them, **trismus**, a condition characterized by a limited ability to open the mouth, is particularly debilitating for patients undergoing head and neck radiotherapy.

Trismus is often a consequence of **muscle atrophy**, fibrosis, or neurological damage induced by the treatment, particularly affecting muscles involved in mastication, such as the **masseter** and **medial pterygoid**.

Muscle atrophy in these regions leads not only to a **functional decline** (chewing difficulty, weight loss) but can also be an early sign of **malnutrition**, affecting overall prognosis and quality of life. Detecting and monitoring such changes is crucial, yet remains a clinical challenge due to the lack of accessible, objective, and automated tools.

In response to this problem, we developed a multidisciplinary project that combines **artificial intelligence**, **medical imaging**, and **nutrition**. The goal was to assist clinicians in identifying early signs of muscle atrophy linked to trismus and to suggest appropriate nutritional interventions.

Leveraging a **deep learning model based on the U-Net architecture**, we successfully achieved automatic segmentation of the masseter and medial pterygoid muscles from medical images acquired before and after radiotherapy. By comparing the segmented surfaces, the system can detect muscle loss, suggest a diagnosis of atrophy, and calculate the BMI using weight and height data.

This analytical process was embedded in a **user-friendly graphical interface** designed for both technicians and physicians. The platform enables : patient information entry, image upload (before/after), automatic segmentation, surface comparison, detection of atrophy, BMI computation, and generation of personalized nutritional recommendations.

It is important to emphasize that **this interface is still under development**. It is currently a functional prototype that showcases the feasibility and potential of the approach, but further improvements and validations are needed before clinical deployment.

By combining clinical insight, algorithmic efficiency, and intuitive design, our application **NutrAdio,** lays the foundation for a complete digital tool to support the care of cancer patients. It bridges the gap between radiotherapy side effect detection and tailored nutritional intervention.

However, this work is not without its **limitations**: dependency on the quality and consistency of medical images, inter-patient anatomical variability, limited dataset for training and validation, current focus on only two masticatory muscles. **Future work** should aim to: expand the training dataset across multiple institutions, enhance the deep learning model's generalizability, include other clinically relevant muscles or tissues, integrate multimodal patient data (clinical, nutritional, biological), and establish a feedback loop between the AI and medical experts.

Eventually, the system could evolve into a **clinical decision-support platform** that not only detects but predicts the risk of complications like trismus, and facilitates interdisciplinary communication between oncologists, radiotherapists, and dietitians. This research opens a promising path at the intersection of **precision medicine** and **AI-powered healthcare.**

Bibliography

[1] J. K. V. E. Seedat, «Epidemiology and demographics of head and neck cancer in Africa,» *African journal of primary health care & family medecine,* p. 15, 2023.

[2] M. G. G. S. A. Gormley, «Reviewing the epidemiology of head and neck cancer: definitions, trends and risk factors,» *Br Dent J 233,* pp. 780-786, 2022.

[3] National Cancer Institute, «Head and neck cancer- patient version [Internet],» 2023. [Online]. [Accessed on April 6, 2025].

[4] L. S. T. A. Yu VX, «Smoking and Head and Neck Cancer,» *JAMA Otolaryngol Head Neck Surg,* p. 470, 2023.

[5] B. Mravec, «Head and Neck Cancer,» *Neurobioogy of cancer. Springer,Cham,* 2024.

[6] A. F. Alfouzan, «Radiation therapy in head and neck cancer,» *Saudi Medical Journal,* pp. 247-254, 2021.

[7] G. M. A. F. S. V. R. O. B. A. J.-F. Daniela Alterio, «Modern radiotherapy for head and neck cancer,» *Seminars in Oncology,* pp. 233-245, 2019.

[8] Santé Sur le Net, «Radiothérapie : une technique du cancer [Internet],» 2023. [Online]. [Accessed on April 7, 2025].

[9] «Prognostic and Therapeutic Potential of Nuclear Receptors in Head and Neck Squamous celle Carcinomas,» [Online]. [Accessed on May 3, 2025].

[10] «Official website of the United States goverment,» 25 05 2021. [Online]. [Accessed on May 3, 2025].

[11] Y. &. W. R. &. B. A. &. Z. X. &. P. Y. &. W. H. Du, «TIGRE-VarianCBCT for on-board cone-beam computed tomography, an open-source toolkit for imaging, dosemetry and clinical research,» *Physica Medica,* pp. 33-45, 2022.

[12] A. &. K. K. Kibrom, «Adaptative radiation therapy for bladder cancer: Areview of adaptative techniques used in clinical practice,» *Journal of Medical Radiation Sciences,* p. 62, 2015.

[13] S. S. C. H. A. C. J.-M. S. C. Chargari, «Traitement des effets tardifs après la radiothérapie : quoi de neuf?,» *Cancer/Radiothérapie,* pp. 602-611, 2020.

[14] J. e. a. Kim, «Neck muscle atrophy and soft-tissue fibrosis after neck dissection and postoperative radiotherapy for oral cancer,» *Radiation oncology journal,* p. 344, 2015.

[15] V. e. a. Dilalla, «Radiotherapy side effects: integrating a survivorship clinical lens to better serve patients,» *Current oncology (Toronto, Ont),* pp. 107-112, 2020.

[16] J. e. a. Du, «Epigenetics of Skeletal Muscle Atrophy,» *International journal of molecular sciences,* 2024.

[17] X. e. a. Wang, «Plants-Derived Treatments for different Types of Muscle Atrophy,» *Phytotherapy research: PTR,* 2025.

[18] Y. S. X. D. F. e. a. Zhou, «Radiation-induced muscle fibrosis rat model: establishment and valuation,» *Radiat Oncol ,* 2018.

[19] S. L. S. Y. D. J. L. R. L. R. B. ,. J. L. R. J. H. H. Alan M Nichol, «Quantification of masticatory muscle atrophy after high-dose radiotherapy,» *International Journal of Radiation Oncology*Biology*Physics,* pp. 1170-1179, 2003.

[20] C.-L. C. J.-B. L. M.-H. W. F.-J. S. Y.-T. J. S.-M. H. Y.-J. C. Jie Lee, «Skeletal Muscle Loss Is an Imaging Biomarker of Outcome after Definitive Chemiotherapy for Locally Advanced Cervical Cancer,» *Clin Cancer ,* pp. 5028-5036, 2018.

[21] Y. &. I. J. &. L. D. &. N. R. &. L. H. &. Y. D. Fu, «Automatic segmentation of CT images for ventral body composition analysis,» *Medical Physics,* p. 47, 2020.

[22] Q. R. X. W. H. Y. C. X. L. F. Xu Y, «Advances in Medical Image Segmentation: A Comprehensive Review of Traditionl, Deep Learning and Hybrid Approaches,» *Bioengineering,* vol. 11, n° %110, p. 1034, 2024.

[23] I. Brook, «Late side effects of radiation treatment for head and neck cancer,» *Radiation oncology journal,* vol. 38, n° %12, pp. 84-92, 2020.

[24] K. e. a. Faravel, «Trismus Occurrence and Link With Radiotherapy Doses in Head and Neck Cancer Patients Treated With Chemiotherapy,» *Integrative cancer therapies,* vol. 22, 2023.

[25] P. e. a. Ramia, «Radiation-Induced Fibrosis in Patients with Head and Neck Cancer: A eview of Pathogenesis and Clinical Outcomes,» *Clinical Medecine Insights.Oncology ,* vol. 16, 2022.

[26] C. J. W. L.-Y. T. Y.-M. S. P.-W. L. L.-J. e. a. Hsieh L-C, «Predicting the severity and Prognosis of Trismus after Intensity-Modulated Radiation Therapyfor Oral Cancer Patients by Magnetic Resonance Imaging,» *PLoS One,* vol. 9, n° %13, 2014.

[27] C. G. C. C. V. C. E. M. C. B. L. Z. F. C. A. T. M. G. P. U. R. G. C. L. Orges Spahiu, «Definitive radiotherapy and trismus in HNC patients: A critical review,» *Critical Reviews in Oncology/Hematology,* vol. 203, 2024.

[28] D. M. C. K. A. V. M. P. F. C. D. J. C. J. Charters E, «Trismus therapy devices: A systematic review,» *Oral Oncology,* 2022.

[29] C. e. a. Lo Nigro, «Head and Neck cancer: improving outcomes with a multidisciplinary approach,» *Cancer management and research ,* vol. 9 , pp. 363-371, 2017.

[30] C. e. a. Torregrosa, «Physical Activity as the Best Supportive Care in Cancer: The Clinician's and the Researcher's Perspectives,» *Cancers,* vol. 14, n° %121, 2022.

[31] IAEA, «Internationnal Atomic Energy Agency,» [Online]. Available: https://www.iaea.org/ [Accessed on May 3, 2025].

[32] X. P. Z. L. Y. e. a. Yang, «Changes in 3-Dimensional Measurements of Masseter Muscle After Orthognathic Surgery in Patients with Facial Asymmetry,» *Aesth Plast Surg,* vol. 48, pp. 3751-3757, 2024.

[33] M. Ferrillo, E. Sommadossi, L. Raciti, D. Calafiore, K. Mezian, V. Tarantino, M. Vecchio, U. Longo, L. Losco et A. de Sire, «The Role of Botulinum Toxin for Masseter Muscle Hypertrophy: A Comprehensive Review,» *Toxins,* vol. 17, n° %191, 2025.

[34] R. M. Jain P, «Anatomy, Head and Neck, Medial Pterygoid Muscle,» 2025. [Online]. [Accessed on May 3, 2025].

[35] E. A. N. P. L. T. C. F. C. O. N. Pettersson, «Decreased Rates of Radiation-induced Trismus and Lowered Mastication Structure Doses in Patients Treated for Head and Neck Cancer During the Last Two Decades,» *Clinical Oncology,* vol. 36, n° %110, pp. 388-397, 2024.

[36] Y. e. a. Shen, «Nutritional Strategies for Muscle Atrophy: Current Evidence and Underlying Mechanisms,» *Molecular nutrition & food research,* vol. 68, n° %110, 2024.

[37] Q. H. K. C. a. Z. W. Tong T, «Role of nutrition in skeletal muscle atrophy and sarcopenia,» *Front.Nutr,* vol. 11, 2024.

[38] TCIA, «The Cancer Imaging Archeive,» [Oonline]. Available: https://cancerimagingarchive.net .

[39] S. F. ResearchGate, «Mitochondrial dysfunction: roles in skeletal muscle atrophy,» [Online]. Available: https://www.researchgate.net/figure/Molecular-mechanism-of-skeletal-muscle-atrophy-Inflammation-and-oxidative-stress-lead-to_fig1_372654063 . [Accessed on May 12, 2025].

[40] S. I. e. a. Muzic, «Reproducibility of manual segmentation in muscle imaging,» *Acta myologica : myopathies and cardiomyopathies : official journal of the Mediterranean Society of Myology ,* vol. 40, n° %13, pp. 116-123, 2021.

[41] S. M. S. S. G. M. G. P. M. P. E. G. C. Marco Penso, «Automated left and right ventricular chamber segmentation in cardiac magnetic resonance images using dense fully convolutional neural network,» *Computer Methods and Programs in Biomedecine,* vol. 204, 2021.

[42] K. &. M. D. &. K. I. &. J. S. &. C. M. Avhad, «Soil Nutrients Analysis Techniques and Crop/ Fertilizers Prediction-A Review,» *Internationnal journal of engineering and Technical Research,* vol. 11, 2022.

[43] G. &. F. A. &. N. B. &. M. R. Pereira, «Active Fire Detection in Landsat-8 Imagery: a Large-Scale Dataset and a Deep-Learning Study,» 2021.

[44] C. a. F. F. a. D. G. a. H. C. C. a. D. A. a. S. S. Williams, «A Unified Framework for U-Net Design and Analysis,» chez *Advances in Neural Information Processing Systems*, A. Oh and T. Naumann and A. Globerson and K. Saenko and M. Hardt and S. Levine, 2023, pp. 27745-27782.

[45] J. Zhuang, «LadderNet: Multi-path networks based on U-Net for medical image segmentation,» *arXiv preprint ,* 2018.

[46] «Radiation Oncology,» *Radiation Oncology Journal,* vol. 38, n° %12, pp. 84-92, 2020.

[47] ResearchGate, «Nutrition interventions to treat low muscle mass in cancer,» Science Figure on ResearchGate. [Online]. [Accessed on May 18, 2025].

[48] Beurtey/CEA, «Le scanner,» [En ligne]. Available: https://www.cea.fr/comprendre/Pages/sante-sciences-du-vivant/essentiel-sur-imagerie-medicale.aspx.

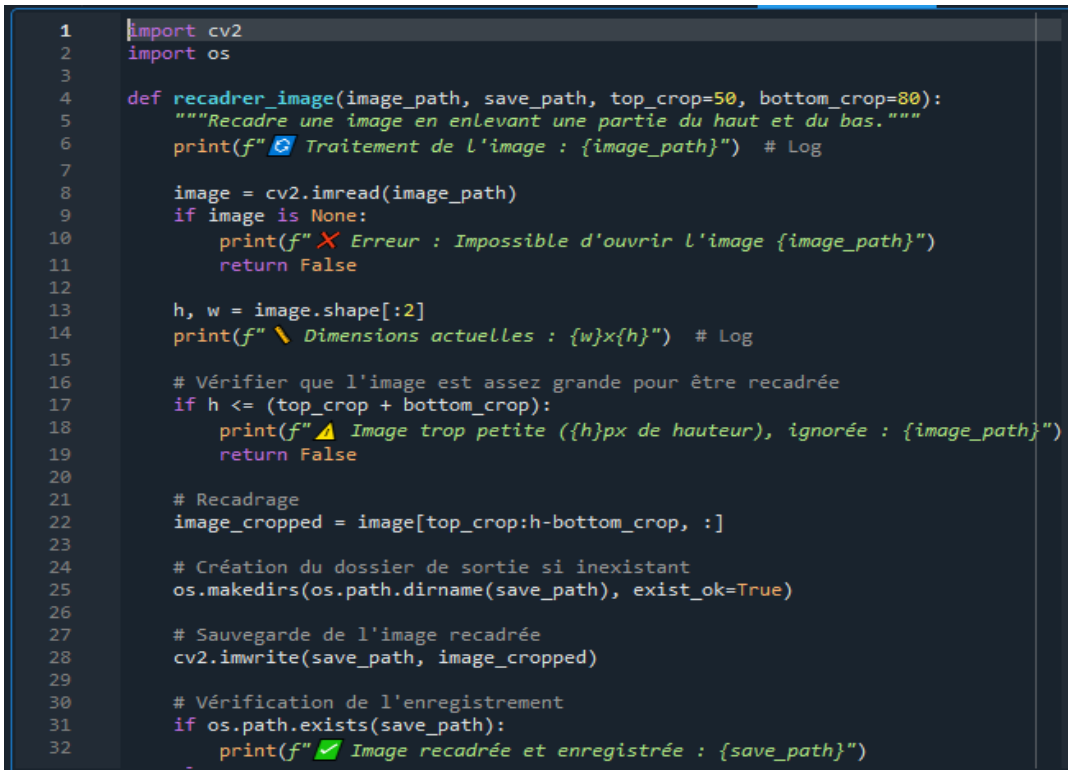
[49] R. B. T. J. V. J. D. T. Y. Z. H. Z. J. W. A. R. S. J. K. X.-M. L. M. M. Kerrie Hayes, «Comparison of manual and semi-automated segmentation methods to evaluate hippocampus volume in APP and PS1 transgenic mice obtained via in vivo magnetic resonance imaging,,» *Journal of Neuroscience Methods,* vol. 221, pp. 103-111, 2014.

[50] G. P. W. C. R. F. S. L. J. C. Quanfa Zhang, «A semi-automatic segmentation procedure for feature extraction in remotely sensed imagery,» *Computers & Geosciences,* vol. 31, pp. 289-296, 2014.

[51] ∗. M. B. 2. T. H. Abdellatif Bouzid-Daho 1, «Segmentation by an Improved Mathematical Morphology Approach,» *Signals and Intelligent Systems Laboratory (LISSI), University UPEC, Creteil, France 2LERICA Laboratory, Badji Mokhtar University, Annaba, Algeria ,* 2020.

Appendices

- **Appendix 1:** Python script for removing image annotations



```python
import cv2
import os

def recadrer_image(image_path, save_path, top_crop=50, bottom_crop=80):
    """Recadre une image en enlevant une partie du haut et du bas."""
    print(f"🔄 Traitement de l'image : {image_path}")  # Log

    image = cv2.imread(image_path)
    if image is None:
        print(f"❌ Erreur : Impossible d'ouvrir l'image {image_path}")
        return False

    h, w = image.shape[:2]
    print(f"📏 Dimensions actuelles : {w}x{h}")  # Log

    # Vérifier que l'image est assez grande pour être recadrée
    if h <= (top_crop + bottom_crop):
        print(f"⚠️ Image trop petite ({h}px de hauteur), ignorée : {image_path}")
        return False

    # Recadrage
    image_cropped = image[top_crop:h-bottom_crop, :]

    # Création du dossier de sortie si inexistant
    os.makedirs(os.path.dirname(save_path), exist_ok=True)

    # Sauvegarde de l'image recadrée
    cv2.imwrite(save_path, image_cropped)

    # Vérification de l'enregistrement
    if os.path.exists(save_path):
        print(f"✅ Image recadrée et enregistrée : {save_path}")
    else:
        print(f"❌ ERREUR : Échec de l'enregistrement de {save_path}")

    return True

def traiter_toutes_les_images(dossier_base, dossier_sortie, top_crop=50, bottom_crop=80):
    """Parcourt tous les patients et leurs images, applique le recadrage et les enregistre."""
    if not os.path.exists(dossier_base):
        print(f"❌ Dossier introuvable : {dossier_base}")
        return

    for patient in os.listdir(dossier_base):
        chemin_patient = os.path.join(dossier_base, patient)
        if not os.path.isdir(chemin_patient):
            print(f"⚠️ Ignoré (ce n'est pas un dossier) : {chemin_patient}")
            continue

        print(f"📂 Traitement du patient : {patient}")

        dossier_patient_sortie = os.path.join(dossier_sortie, patient)
        os.makedirs(dossier_patient_sortie, exist_ok=True)

        for fichier in os.listdir(chemin_patient):
            if not fichier.lower().endswith(('.png', '.jpg', '.jpeg')):
                print(f"⚠️ Fichier ignoré (non-image) : {fichier}")
                continue

            chemin_image = os.path.join(chemin_patient, fichier)
            chemin_image_sortie = os.path.join(dossier_patient_sortie, fichier)

            # Appliquer le recadrage
            recadrer_image(chemin_image, chemin_image_sortie, top_crop, bottom_crop)

# Définition des chemins
dossier_base = r"C:\Users\INFOLUX\Documents\NutrAdio\images_a_annoter"
dossier_sortie = r"C:\Users\INFOLUX\Documents\NutrAdio\images_annotees"

# Lancement du traitement
traiter_toutes_les_images(dossier_base, dossier_sortie, top_crop=80, bottom_crop=90)
```

➢ **Appendix 2:** Python script for overlaying masks onto images

```python
import os
import SimpleITK as sitk
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Chemins des dossiers
base_dossier_images = r"C:\Users\INFOLUX\Documents\NutrAdio\images_redim"
base_dossier_masks = r"C:\Users\INFOLUX\Documents\NutrAdio\masks"

# Liste des patients sélectionnés
patients = ["HNSCC-01-0053", "HNSCC-01-0055", "HNSCC-01-0058", "HNSCC-01-0059", "HNSCC-01-00101"]

# Boucle sur chaque patient
for patient in patients:
    dossier_images = os.path.join(base_dossier_images, patient)  # plus de "images"
    dossier_masks = os.path.join(base_dossier_masks, patient)    # plus de "masks"

    # Vérifier si les dossiers existent
    if not os.path.exists(dossier_images) or not os.path.exists(dossier_masks):
        print(f"⚠ Dossier manquant pour {patient}")
        continue

    # Associer images et masques
    correspondances = {
        "apres_neck_muscle.jpg": "apres_neck_muscle.seg.nrrd",
        "avant_neck_muscle.jpg": "avant_neck_muscle.seg.nrrd"
    }

    # Boucle sur les fichiers correspondants
    for img_name, mask_name in correspondances.items():
        img_path = os.path.join(dossier_images, img_name)
        mask_path = os.path.join(dossier_masks, mask_name)

```

```python
        # Vérifier si les fichiers existent
        if not os.path.exists(img_path):
            print(f"⚠ Image non trouvée pour {patient}: {img_name}")
            continue
        if not os.path.exists(mask_path):
            print(f"⚠ Masque non trouvé pour {patient}: {mask_name}")
            continue

        # Charger l'image JPG
        image = cv2.imread(img_path)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

        # Charger le masque NRRD
        mask_nrrd = sitk.ReadImage(mask_path)
        mask = sitk.GetArrayFromImage(mask_nrrd)[0]

        # Redimensionner le masque à la taille de l'image
        mask = cv2.resize(mask, (image.shape[1], image.shape[0]), interpolation=cv2.INTER_NEAREST)

        # Créer une image superposée (rouge pour le masque)
        mask_overlay = np.zeros_like(image)
        mask_overlay[:, :, 0] = mask * 255  # Rouge
        mask_overlay[:, :, 1] = mask * 0    # Pas de vert
        mask_overlay[:, :, 2] = mask * 0    # Pas de bleu

        # Superposer avec transparence
        alpha = 0.5
        superposed = cv2.addWeighted(image, 1, mask_overlay, alpha, 0)

        # Afficher l'image et le masque
        plt.figure(figsize=(10, 5))
        plt.subplot(1, 2, 1)
        plt.imshow(image)
        plt.title(f"{patient} - Image originale: {img_name}")
        plt.axis("off")

        plt.subplot(1, 2, 2)
        plt.imshow(superposed)
        plt.title(f"{patient} - Superposition: {img_name}")
        plt.axis("off")
```

- **Appendix 3:** Python script for data augmentation.

```python
import os
import cv2
import numpy as np
import random
import SimpleITK as sitk

# Paramètres
patient_id = "data"
image_patient_dir = f'C:/Users/INFOLUX/Documents/NutrAdio/tout/images/{patient_id}'
mask_patient_dir = f'C:/Users/INFOLUX/Documents/NutrAdio/tout/masks/{patient_id}'
output_image_dir = f'C:/Users/INFOLUX/Documents/NutrAdio/data_augmentation/images/{patient_id}'
output_mask_dir = f'C:/Users/INFOLUX/Documents/NutrAdio/data_augmentation/masks/{patient_id}'

os.makedirs(output_image_dir, exist_ok=True)
os.makedirs(output_mask_dir, exist_ok=True)

# Transformations
def rotate_90(image, mask): return np.rot90(image), np.rot90(mask)
def rotate_180(image, mask): return np.rot90(image, 2), np.rot90(mask, 2)
def rotate_270(image, mask): return np.rot90(image, 3), np.rot90(mask, 3)
def change_contrast(image, mask):
    alpha = random.uniform(0.8, 1.2)
    beta = random.randint(-30, 30)
    return cv2.convertScaleAbs(image, alpha=alpha, beta=beta), mask
def change_brightness(image, mask):
    beta = random.randint(-100, 100)
    return cv2.convertScaleAbs(image, alpha=1.0, beta=beta), mask
def zoom(image, mask):
    h, w = image.shape[:2]
    scale = random.uniform(0.8, 1.2)
    new_h, new_w = int(h * scale), int(w * scale)
    image_zoomed = cv2.resize(image, (new_w, new_h), interpolation=cv2.INTER_LINEAR)
    mask_zoomed = cv2.resize(mask, (new_w, new_h), interpolation=cv2.INTER_NEAREST)
    top = max((h - new_h) // 2, 0)
    bottom = max(h - new_h - top, 0)
    left = max((w - new_w) // 2, 0)
    right = max(w - new_w - left, 0)
    image_padded = cv2.copyMakeBorder(image_zoomed, top, bottom, left, right, cv2.BORDER_CONSTANT, value=0)
    mask_padded = cv2.copyMakeBorder(mask_zoomed, top, bottom, left, right, cv2.BORDER_CONSTANT, value=0)
    image_resized = cv2.resize(image_padded, (w, h), interpolation=cv2.INTER_LINEAR)
    mask_resized = cv2.resize(mask_padded, (w, h), interpolation=cv2.INTER_NEAREST)
    return image_resized, mask_resized
def add_gaussian_noise(image, mask):
    noise = np.random.normal(0, 5, image.shape).astype(np.uint8)
    noisy_image = cv2.add(image, noise)
    return noisy_image, mask
def translate(image, mask):
    h, w = image.shape[:2]
    tx, ty = random.randint(-20, 20), random.randint(-20, 20)
    M = np.float32([[1, 0, tx], [0, 1, ty]])
    image_translated = cv2.warpAffine(image, M, (w, h), borderMode=cv2.BORDER_CONSTANT, borderValue=0)
    mask_translated = cv2.warpAffine(mask, M, (w, h), flags=cv2.INTER_NEAREST, borderMode=cv2.BORDER_CONSTANT, borderValue=0)
    return image_translated, mask_translated

transformations = [rotate_90, rotate_180, rotate_270, change_contrast, change_brightness, zoom, add_gaussian_noise, translate]

# Lecture des fichiers
image_files = sorted([f for f in os.listdir(image_patient_dir) if f.lower().endswith('.jpg')])
mask_files = sorted([f for f in os.listdir(mask_patient_dir) if f.lower().endswith('.nrrd')])

print(f"[⚠] {len(image_files)} images et {len(mask_files)} masques détectés.")

counter = 1
for img_file, mask_file in zip(image_files, mask_files):
    image_path = os.path.join(image_patient_dir, img_file)
    mask_path = os.path.join(mask_patient_dir, mask_file)
```

```
    image = cv2.imread(image_path)
    if image is None:
        print(f"[X] Image introuvable : {image_path}")
        continue

    try:
        mask_itk = sitk.ReadImage(mask_path)
        mask = sitk.GetArrayFromImage(mask_itk)[0]  # Supposer une seule slice
    except Exception as e:
        print(f"[X] Erreur de lecture du masque {mask_path} : {e}")
        continue

    for transform in transformations:
        aug_image, aug_mask = transform(image, mask)

        # Redimensionner si nécessaire
        if aug_mask.shape != mask.shape:
            aug_mask = cv2.resize(aug_mask, (mask.shape[1], mask.shape[0]), interpolation=cv2.INTER_NEAREST)

        # Sauvegarde image augmentée
        img_name = f"neck-{counter:03d}.jpg"
        cv2.imwrite(os.path.join(output_image_dir, img_name), aug_image)

        # Sauvegarde masque augmentée avec SimpleITK
        try:
            aug_mask_itk = sitk.GetImageFromArray(np.expand_dims(aug_mask.astype(np.uint8), axis=0))  # ajouter la 3e dimension
            aug_mask_itk.CopyInformation(mask_itk)
            mask_name = f"mask-{counter:03d}.nrrd"
            sitk.WriteImage(aug_mask_itk, os.path.join(output_mask_dir, mask_name))
        except Exception as e:
            print(f"[X] Erreur d'écriture du masque {mask_name} : {e}")
            continue
```

- Alpha (contrast scale): A multiplicative factor that adjusts the contrast of the image. Values >1 increase contrast, values <1 reduce it.
- Beta (brightness shift): An additive value that brightens (positive) or darkens (negative) the image.
- Sigma (standard deviation of noise): Used in Gaussian noise, controls the intensity of the random variations added to pixel values.