People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Kasdi Merbah University – Ouargla

Faculty of Mathematics and Matter Sciences

Department of Physics

Master's Thesis

Submitted in partial fulfilment of the requirements
for the degree of **Master in Medical Physics**

Deep Learning-Based Automatic Segmentation of Gross Tumor Volume in Non-Small Cell Lung Cancer Patients

---

Presented by:

BEN NANA Roumissa and DOUHA Marwa

Examination Committee

Dr. Ayat Zahia	President	Kasdi Merbah University – Ouargla
Dr. Mokaddem Zakaria	Examiner	Kasdi Merbah University – Ouargla
Pr. Bentouila Omar	Supervisor	Kasdi Merbah University – Ouargla

Academic Year 2025–2026

Dedication

We dedicate this work to:

Our mothers and fathers, and to our wonderful families and our dear beloved ones, we would like to express our deepest gratitude and appreciation to all of you. You have always been our unmatched support and the light that brightens our path toward success.

You have given us the strength and courage to pursue and achieve our dreams.

Thank you all for the great trust you have placed in us and for every moment of your support and encouragement. Without you, we would not have been able to accomplish all of this.

You are the stars that illuminate the sky of our life, and we will always hold the deepest respect and admiration for you.

Acknowledgments

Firstly, thanks go to **Allah** before and after, who enabled us to complete this work.

We would like to express our deep gratitude to **Prof. Bentouila Omar**, for his constructive criticism, help, guidance, and patience. We could not have completed our thesis without his enlightening instruction.

Special thanks to **Dr. Ayat Zahia** for agreeing to be the president of the jury for this thesis.

We also thank **Dr. Mokaddem Zakaria** for being one of our jury members.

We would also like to thank all our friends at the Laboratory for the Development of New and Renewable Energies in Arid Zones, where we conducted this work.

We are sincerely grateful to our family members. Thank you for consistently encouraging us to be brave, independent, and optimistic. We appreciate their support over the years.

Last but not least, we express our thanks to all the other **professors**, **researchers**, and **secretaries** who have helped us in the past years.

Abstract

Accurate delineation of the Gross Tumor Volume (GTV) is a critical step in radiotherapy planning for Non-Small Cell Lung Cancer (NSCLC), yet manual contouring by radiation oncologists is time-consuming and subject to significant inter-observer variability. This thesis presents an automated GTV segmentation pipeline based on a three-dimensional U-Net (3D U-Net) architecture with 5.6 million parameters, trained and evaluated on 65 patients from the publicly available NSCLC-Radiomics dataset (The Cancer Imaging Archive). After preprocessing (Hounsfield unit normalization and tumor-centered cropping to $96 \times 192 \times 192$ voxels), the network was trained with a combined Dice and Weighted Binary Cross-Entropy loss (positive class weight = 100) to address the severe class imbalance inherent in tumor segmentation, on a single NVIDIA Tesla T4 GPU (Google Colab) in approximately 11 minutes. On an independent test set of 10 patients, the model achieved a mean Dice Similarity Coefficient (DSC) of 0.547 ± 0.228 (best individual result 0.856). A strong positive correlation was observed between tumor volume and accuracy: large tumors (≥ 100 cm^3) reached a mean DSC of 0.750, comparable to published results, while small tumors (< 20 cm^3) yielded only 0.259. The model exhibited high Sensitivity (0.855) relative to Precision (0.474), favoring over-segmentation—a clinically preferable behavior in radiotherapy planning. A Test Time Augmentation experiment with spatial flipping produced a negative result (DSC decreased to 0.538), attributed to the disruption of anatomical laterality cues in the thorax. These results demonstrate the feasibility of training a competitive GTV segmentation model under limited data and computational constraints, and contribute a detailed size–performance analysis often reported only in aggregate in the literature.

Keywords: Lung cancer, NSCLC, Gross Tumor Volume, automatic segmentation, deep learning, 3D U-Net, Dice Similarity Coefficient, radiotherapy planning, TCIA, class imbalance.

Résumé

La délimitation précise du Volume Tumoral Macroscopique (GTV) constitue une étape critique dans la planification de la radiothérapie du Cancer Bronchique Non à Petites Cellules (CBNPC), or le contourage manuel par les radio-oncologues est chronophage et sujet à une variabilité inter-observateur significative. Ce mémoire présente un pipeline automatisé de segmentation du GTV basé sur une architecture U-Net tridimensionnelle (3D U-Net) comportant 5,6 millions de paramètres, entraîné et évalué sur 65 patients du jeu de données public NSCLC-Radiomics (The Cancer Imaging Archive). Après prétraitement (normalisation des unités Hounsfield et recadrage centré sur la tumeur en $96 \times 192 \times 192$ voxels), le réseau a été entraîné avec une fonction de perte combinée (Dice + Entropie Croisée Binaire Pondérée, poids de la classe positive = 100) afin de compenser le déséquilibre de classes sévère inhérent à la segmentation tumorale, sur un seul GPU NVIDIA Tesla T4 (Google Colab) en environ 11 minutes. Sur un ensemble de test indépendant de 10 patients, le modèle a obtenu un coefficient de similarité de Dice (DSC) moyen de 0.547 ± 0.228 (meilleur résultat individuel 0.856). Une corrélation positive marquée a été observée entre le volume tumoral et la précision : les tumeurs volumineuses (≥ 100 cm^3) ont atteint un DSC moyen de 0.750, comparable aux résultats publiés, tandis que les petites tumeurs (< 20 cm^3) n'ont obtenu que 0.259. Le modèle a présenté une Sensibilité élevée (0.855) par rapport à la Précision (0.474), favorisant la sur-segmentation — comportement cliniquement préférable en radiothérapie. Une expérience d'augmentation au moment du test (TTA) par retournement spatial a donné un résultat négatif (DSC réduit à 0.538), attribué à la perturbation des repères de latéralité dans le thorax. Ces résultats démontrent la faisabilité d'entraîner un modèle compétitif avec des données et des ressources limitées, et fournissent une analyse détaillée de la relation taille-performance rarement documentée.

Mots-clés : Cancer du poumon, CBNPC, Volume Tumoral Macroscopique, segmentation automatique, apprentissage profond, U-Net 3D, coefficient de similarité de Dice, planification de radiothérapie, TCIA, déséquilibre de classes.

ملخص

يُعد التحديد الدقيق لحجم الورم الجسيم (GTV) خطوة حاسمة في تخطيط العلاج الإشعاعي لسرطان الرئة ذي الخلايا غير الصغيرة (NSCLC).يتطلب الرسم اليدوي من قبل أطباء الأورام الإشعاعيين وقتاً طويلاً (30 إلى 90 دقيقة لكل مريض) ويخضع لتباين كبير بين المراقبين. تقدم هذه المذكرة نظام تحديد آلي لحجم الورم الجسيم يعتمد على بنية U-Net ثلاثية الأبعاد(3D U-Net) ، تم تدريبه وتقييمه على قاعدة بيانات NSCLC-Radiomics المتاحة على أرشيف التصوير السرطاني.(TCIA)

تم استخدام مجموعة مكونة من 65 مريضاً مصاباً بسرطان الرئة ذي الخلايا غير الصغيرة، مع صور مقطعية محوسبة (CT) وتحديدات يدوية للورم. بعد المعالجة المسبقة (تطبيع وحدات هاونسفيلد، القص المركّز حول الورم بأبعاد 96×192×192 فوكسل)، تم تدريب شبكة 3 D U-Netتحتوي على 5.6 مليون معامل باستخدام دالة خسارة مركبة + Dice) الانتروبيا التبادلية الثنائية الموزونة، وزن الفئة الإيجابية = 100 (لمعالجة اختلال التوازن الشديد بين الفئات. أُجري التدريب على وحدة معالجة رسومية NVIDIA Tesla T4 (Google Colab)واكتمل في حوالي 11 دقيقة.

على مجموعة اختبار مستقلة مكونة من 10 مرضى، حقق النموذج معامل تشابه (DSC) Dice متوسط قدره 0.228 ± 0.547، مع أفضل نتيجة فردية بلغت 0.856. لوحظ ارتباط إيجابي قوي بين حجم الورم ودقة التجزئة: حققت الأورام الكبيرة (≥100 سم³) معامل DSCمتوسط قدره 0.750 مقارب للنتائج المنشورة في الأدبيات، بينما حققت الأورام الصغيرة (<20 سم³) قيمة 0.259. أظهر النموذج حساسية عالية (0.855) مقارنة بالدقة (0.474)، مفضلاً التوسع في التحديد على نقصانه — وهو سلوك مفضّل سريرياً في سياق تخطيط العلاج الإشعاعي.

أظهرت تجربة تعزيز وقت الاختبار (TTA) بالانعكاس المكاني نتيجة سلبية) انخفض DSC المتوسط إلى 0.538(، وذلك بسبب تعطيل إشارات التماثل التشريحي في الصدر.

تُثبت هذه النتائج إمكانية تدريب نموذج تنافسي لتحديد حجم الورم الجسيم ببيانات وموارد حاسوبية محدودة، وتوفر تحليلاً مفصلاً للعلاقة بين الحجم والأداء نادراً ما يُوثّق في الأدبيات.

الكلمات المفتاحية :سرطان الرئة، NSCLC، حجم الورم الجسيم، التجزئة الآلية، التعلم العميق، U-Netثلاثي الأبعاد، معامل تشابه Dice، تخطيط العلاج الإشعاعي، TCIA، اختلال توازن الفئات.

Contents

Dedication i

Acknowledgments ii

Abstract iii

Résumé iv

List of Abbreviations xii

General Introduction 1

1 Background and Literature Review 4

1.1 Lung Cancer . . . 4

1.1.1 Epidemiology . . . 4

1.1.2 Classification of Lung Cancer . . . 5

1.1.3 Staging . . . 5

1.1.4 Treatment Modalities . . . 6

1.2 Target Volume Delineation in Radiotherapy . . . 6

1.2.1 The Radiotherapy Planning Process . . . 6

1.2.2 ICRU Target Volume Definitions . . . 7

1.2.3 The Challenge of Manual GTV Delineation . . . 8

1.3 Medical Imaging Fundamentals . . . 8

1.3.1 Computed Tomography (CT) . . . 8

1.3.2 DICOM Standard . . . 9

1.4 Deep Learning for Medical Image Segmentation . . . 10

1.4.1 From Machine Learning to Deep Learning . . . 10

1.4.2 Convolutional Neural Networks (CNN) . . . 10

1.4.3 Image Segmentation . . . 11

1.4.4 The U-Net Architecture . . . 12

1.5 State of the Art in Automatic GTV Segmentation . . . 14

1.5.1 Early Approaches . . . 14

1.5.2 Deep Learning Approaches . . . 14
1.5.3 Key Observations from the Literature . . . 16
1.5.4 Positioning of This Work . . . 17

2 Materials and Methods 18
2.1 Overview of the Proposed Pipeline . . . 18
2.2 Dataset . . . 19
2.2.1 NSCLC-Radiomics Collection . . . 19
2.2.2 Patient Selection . . . 19
2.2.3 Tumor Volume Characteristics . . . 19
2.3 Data Acquisition and Preprocessing . . . 20
2.3.1 Data Acquisition from TCIA . . . 20
2.3.2 CT Volume Extraction . . . 20
2.3.3 GTV Mask Extraction from RTSTRUCT . . . 21
2.3.4 Intensity Normalization . . . 21
2.3.5 Tumor-Centered Cropping . . . 21
2.3.6 Final Dataset Statistics . . . 22
2.4 Network Architecture . . . 22
2.4.1 Design Rationale . . . 22
2.4.2 Overall Architecture . . . 23
2.4.3 Building Blocks . . . 24
2.4.4 Encoder Path . . . 24
2.4.5 Decoder Path with Skip Connections . . . 24
2.4.6 Output Layer . . . 25
2.4.7 Architectural Details . . . 25
2.5 Training Configuration . . . 25
2.5.1 Dataset Splitting . . . 25
2.5.2 Data Augmentation . . . 26
2.5.3 Loss Function . . . 26
2.5.4 Optimizer and Learning Rate Scheduling . . . 28
2.5.5 Training Stability Techniques . . . 28
2.5.6 Early Stopping and Model Selection . . . 28
2.5.7 Hyperparameter Summary . . . 28
2.5.8 Implementation Environment . . . 29
2.6 Evaluation Methodology . . . 29
2.6.1 Inference Pipeline . . . 29
2.6.2 Quantitative Evaluation Metrics . . . 30
2.6.3 Volume-Based Stratification . . . 31
2.6.4 Test Time Augmentation Experiment . . . 31

2.6.5 Reproducibility Considerations . . . 32

3 Results and Discussion **33**
3.1 Training Behavior . . . 33
3.1.1 Loss Convergence . . . 33
3.1.2 DSC Evolution . . . 34
3.1.3 Learning Rate Adaptation . . . 34
3.2 Quantitative Evaluation on the Test Set . . . 34
3.2.1 Summary Statistics . . . 35
3.3 Qualitative Results . . . 36
3.4 Performance Stratification by Tumor Volume . . . 37
3.5 Discussion . . . 38
3.5.1 Comparison with the State of the Art . . . 38
3.5.2 The Size–Performance Relationship . . . 40
3.5.3 Sensitivity–Precision Trade-Off . . . 40
3.5.4 Test Time Augmentation: A Negative Result . . . 41
3.5.5 Limitations . . . 41
3.5.6 Clinical Relevance and Potential Applications . . . 42

Conclusion and Future Work **43**

References **46**

A Supplementary Materials **50**
A.1 Key Code Excerpts . . . 50
A.1.1 Data Preprocessing . . . 50
A.1.2 3D U-Net Architecture . . . 51
A.1.3 Loss Function . . . 52
A.2 Dataset Details . . . 53
A.2.1 Complete Patient List . . . 53
A.2.2 ROI Names in RTSTRUCT . . . 53
A.3 Additional Results . . . 54
A.3.1 Distribution of Evaluation Metrics . . . 54
A.3.2 Test Time Augmentation: Detailed Comparison . . . 54
A.3.3 Training Log Summary . . . 55
A.3.4 Software Versions . . . 55

List of Figures

1.1 ICRU target volume definitions. The GTV represents the visible tumor extent and is the segmentation target in this thesis. The CTV and PTV include additional margins for microscopic disease and geometric uncertainties, respectively (ICRU Reports 50 and 62). 7
1.2 Schematic illustration of a CNN feature extraction pipeline. Successive convolutional layers extract increasingly abstract features, while pooling layers reduce spatial dimensions. 11
1.3 Conceptual illustration of the U-Net architecture showing the encoder (contracting) path, bottleneck, decoder (expanding) path, and skip connections that preserve spatial details [5]. 12

2.1 Overview of the proposed pipeline for automatic GTV segmentation, from data acquisition to evaluation. 18
2.2 Data preprocessing pipeline showing the five main steps: DICOM extraction, HU normalization, tumor-centered cropping, and storage in compressed NumPy format. 20
2.3 Detailed architecture of the proposed 3D U-Net. The encoder (blue) progressively reduces spatial dimensions while increasing channels (16→256). The decoder (green) restores resolution via transposed convolutions. Skip connections (yellow dashed) concatenate encoder features with decoder features at each level. 23

3.1 Training history over 40 epochs. (a) Training and validation loss convergence. (b) DSC evolution, with the green star indicating the best validation DSC of 0.700 at epoch 30. (c) Learning rate schedule showing automatic reductions at epochs 14 and 36. 33
3.2 Qualitative segmentation results for three representative test cases. Top row: LUNG1-018 (large tumor, DSC = 0.856); middle row: LUNG1-047 (medium tumor, DSC = 0.674); bottom row: LUNG1-062 (small tumor, DSC = 0.311). Green contours: ground truth; red contours: prediction. The rightmost column shows the overlay with quantitative metrics. . . 36

3.3 DSC as a function of ground truth tumor volume. Each point represents one test patient. Colors indicate tumor size categories: red (small, $< 20\,\mathrm{cm}^3$), orange (medium, 20–$100\,\mathrm{cm}^3$), green (large, $\geq 100\,\mathrm{cm}^3$). Dashed lines show the mean DSC within each category. 37

A.1 Box plot distribution of segmentation metrics on the test set: DSC, IoU, Sensitivity, Precision, and HD95. 54

List of Tables

1.1 Typical Hounsfield Unit values for thoracic tissues. 9
1.2 Summary of representative deep learning-based studies for lung tumor / GTV segmentation. Reported DSC values correspond to each study's primary test result as stated in the cited publication. 15

2.1 Summary statistics of the preprocessed dataset. 22
2.2 Layer-by-layer specification of the proposed 3D U-Net. 26
2.3 Summary of training hyperparameters. 29

3.1 Per-patient segmentation results on the test set. GT Vol. and Pred. Vol. denote the ground truth and predicted tumor volumes, respectively. Best results per metric are shown in **bold**. 35
3.2 Summary of segmentation metrics on the test set (10 patients). Values are reported as Mean ± Std (range: Min–Max). 35
3.3 Segmentation performance stratified by tumor volume. 37
3.4 Comparison of this work with representative published lung tumor / GTV segmentation studies. DSC values correspond to each study's reported test result; note differences in training set size, imaging modality, and architecture. 39

A.1 Complete list of patients with GTV volume and split assignment. Volumes are reported in voxels (after cropping to $96 \times 192 \times 192$). 53
A.2 Frequency of ROI names across the 66 patients with RTSTRUCT data. 54
A.3 Per-patient comparison of DSC with and without TTA. ΔDSC = DSC(TTA) − DSC(standard). Negative values indicate degradation. 55
A.4 Key milestones during the training process (40 epochs). 55
A.5 Software and library versions used. 56

List of Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
AMP	Automatic Mixed Precision
BCE	Binary Cross-Entropy
CNN	Convolutional Neural Network
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
DSC	Dice Similarity Coefficient
GPU	Graphics Processing Unit
GTV	Gross Tumor Volume
HD95	Hausdorff Distance (95th percentile)
HU	Hounsfield Unit
IoU	Intersection over Union
ML	Machine Learning
NSCLC	Non-Small Cell Lung Cancer
OAR	Organ At Risk
PTV	Planning Target Volume
RT	Radiotherapy
RTDOSE	Radiation Therapy Dose
RTSTRUCT	Radiation Therapy Structure Set
TCIA	The Cancer Imaging Archive
TTA	Test Time Augmentation

General Introduction

Context and Motivation

Lung cancer is the leading cause of cancer-related death worldwide, responsible for approximately 1.8 million deaths in 2022 [1]. Non-Small Cell Lung Cancer (NSCLC), which accounts for roughly 85 % of all lung cancer cases [2], is frequently treated with radiotherapy, either as a primary modality or in combination with surgery and chemotherapy.

A critical step in radiotherapy planning is the delineation of the Gross Tumor Volume (GTV)—the visible extent of the tumor on diagnostic imaging. This task is performed manually by radiation oncologists on computed tomography (CT) scans, a process that is both time-consuming and labor-intensive [3]. Moreover, manual delineation is inherently subjective: significant inter-observer variability has been reported, with a mean Dice Similarity Coefficient of approximately 0.56 for poorly-defined lung tumors delineated on CT alone [4].

These limitations—time consumption and operator dependence—motivate the development of automated tools capable of producing accurate, reproducible GTV delineations in a fraction of the time required for manual contouring. Deep learning, and in particular the U-Net architecture family [5], [6], has emerged as the most promising approach for this task, achieving state-of-the-art performance in several medical image segmentation benchmarks [7].

Problem Statement

Despite the encouraging results reported in the literature, several practical challenges remain:

- Most published studies rely on large training datasets (200–400 patients) and high-performance computing infrastructure (multi-GPU servers), which may not be available in all research settings.
- The relationship between tumor size and segmentation accuracy is often reported

only in aggregate metrics, obscuring important clinical nuances regarding the reliability of automated contours for different tumor presentations.

- The transferability of models trained on public datasets to local clinical environments remains insufficiently studied [8], particularly in developing countries where imaging equipment and patient demographics may differ from those in Western academic centers.

Objectives

The primary objective of this thesis is to develop and evaluate a deep learning-based pipeline for the automatic segmentation of GTV in NSCLC patients, with a specific emphasis on:

1. Designing a complete and reproducible pipeline—from publicly available DICOM data to quantitative evaluation—that can operate within the computational constraints of freely available cloud resources (Google Colab).

2. Training and evaluating a 3D U-Net architecture on the NSCLC-Radiomics dataset [9] from The Cancer Imaging Archive (TCIA) [10], using clinically relevant evaluation metrics including the Dice Similarity Coefficient, Hausdorff Distance, Sensitivity, and Precision.

3. Investigating the influence of tumor volume on segmentation performance through a stratified analysis, providing insights into the clinical scenarios where the model is most and least reliable.

4. Evaluating the potential of Test Time Augmentation as a post-processing technique for improving segmentation quality.

Thesis Organization

This thesis is organized into three chapters:

Chapter 1: Background and Literature Review. This chapter provides the necessary theoretical foundations, covering lung cancer biology and treatment, medical imaging principles, deep learning concepts, and a review of the existing literature on automatic GTV segmentation.

Chapter 2: Materials and Methods. This chapter describes the complete methodology, including the dataset, preprocessing pipeline, network architecture, training configuration, and evaluation metrics.

Chapter 3: Results and Discussion. This chapter presents the experimental results on the independent test set, provides a detailed analysis of performance stratified by tumor size, and discusses the findings in the context of the existing literature, including a comparison with state-of-the-art methods and an analysis of limitations.

The thesis concludes with a summary of contributions and directions for future research.

Chapter 1

Background and Literature Review

This chapter provides the theoretical foundation for the work presented in this thesis. The chapter begins with an overview of lung cancer and its treatment (Section 1.1), followed by a discussion of target volume delineation in radiotherapy (Section 1.2). Section 1.3 introduces the relevant medical imaging concepts. Section 1.4 covers the deep learning methods used in this work, and Section 1.5 reviews the state of the art in automatic GTV segmentation.

1.1 Lung Cancer

1.1.1 Epidemiology

Lung cancer remains the leading cause of cancer-related mortality worldwide. According to the GLOBOCAN 2022 estimates produced by the International Agency for Research on Cancer (IARC), lung cancer was the most frequently diagnosed cancer that year, accounting for approximately 12.4 % of all new cancer cases globally (close to 2.5 million cases), and was responsible for an estimated 1.8 million deaths, representing 18.7 % of all cancer deaths [1].

In Algeria, lung cancer represents a major public health burden. According to GLOBOCAN 2022 estimates, it is the most frequently diagnosed cancer among Algerian males (accounting for approximately 14.7 % of new cancer cases in men) and the second leading cause of cancer death for both sexes combined [11]. The first national lung cancer registry in Algeria (LuCaReAl) reported that most patients are diagnosed at an advanced stage, with a marked male predominance and tobacco smoking as the primary risk factor [12]. The increasing burden of lung cancer in Algeria and other developing countries underscores the need for efficient diagnostic and therapeutic tools that can operate within the constraints of limited healthcare resources.

1.1.2 Classification of Lung Cancer

Lung cancer is broadly classified into two histological categories [2]:

- **Small Cell Lung Cancer (SCLC):** Accounting for approximately 15 % of all lung cancers, SCLC is characterized by rapid growth and early metastasis. It is strongly associated with smoking and is typically treated with chemotherapy and radiation.
- **Non-Small Cell Lung Cancer (NSCLC):** Representing approximately 85 % of all lung cancers, NSCLC is further divided into three major subtypes:
  - *Adenocarcinoma* (∼40 %): The most common subtype, often found in the peripheral regions of the lung.
  - *Squamous cell carcinoma* (∼30 %): Typically located centrally, near the main bronchi.
  - *Large cell carcinoma* (∼10 %): A heterogeneous group with variable prognosis.

This thesis focuses exclusively on **NSCLC**, which is the subject of the NSCLC-Radiomics dataset used in this study.

1.1.3 Staging

The staging of NSCLC follows the TNM classification system, currently in its eighth edition [13]:

- **T** (Tumor): Describes the size and extent of the primary tumor, ranging from T1 (≤ 3 cm) to T4 (invasion of critical structures).
- **N** (Nodes): Indicates the involvement of regional lymph nodes, from N0 (no involvement) to N3 (contralateral or distant nodal involvement).
- **M** (Metastasis): Indicates the presence (M1) or absence (M0) of distant metastases.

The combination of T, N, and M categories determines the overall stage (I through IV), which guides treatment decisions [13]. Early-stage NSCLC (stages I–II) is primarily treated with surgery, while locally advanced disease (stage III) typically requires a multimodal approach combining radiotherapy and chemotherapy [14].

1.1.4 Treatment Modalities

The principal treatment options for NSCLC include [2], [14]:

1. **Surgery:** Preferred for early-stage disease (stages I–II) when the patient is medically operable. Options include lobectomy, pneumonectomy, and wedge resection.
2. **Radiotherapy (RT):** Used as a primary treatment for patients who are not surgical candidates, and as an adjuvant or neoadjuvant therapy in combination with chemotherapy for locally advanced disease. Modern techniques include three-dimensional conformal radiation therapy (3D-CRT), intensity-modulated radiation therapy (IMRT), and stereotactic body radiation therapy (SBRT).
3. **Chemotherapy:** Systemic treatment using cytotoxic agents, often combined with radiotherapy (chemoradiotherapy) for stage III disease.
4. **Immunotherapy and targeted therapy:** Newer approaches that exploit specific molecular characteristics of the tumor, increasingly used in advanced-stage NSCLC.

1.2 Target Volume Delineation in Radiotherapy

1.2.1 The Radiotherapy Planning Process

Radiotherapy aims to deliver a precisely targeted dose of ionizing radiation to the tumor while minimizing damage to surrounding healthy tissues. The planning process involves several sequential steps:

1. **Imaging:** Acquisition of a planning CT scan in the treatment position.
2. **Contouring:** Manual delineation of the tumor volume and organs at risk (OARs) by the radiation oncologist.
3. **Treatment planning:** Computation of the optimal beam configuration and dose distribution by the medical physicist using a Treatment Planning System (TPS).
4. **Quality assurance:** Verification of the plan before treatment delivery.
5. **Treatment delivery:** Fractionated irradiation over several weeks.

Among these steps, **contouring** (step 2) is widely recognized as one of the most time-consuming and operator-dependent stages, making it a critical bottleneck in the radiotherapy workflow [3].

1.2.2 ICRU Target Volume Definitions

The International Commission on Radiation Units and Measurements (ICRU) has established a standardized nomenclature for target volumes in radiotherapy, defined in ICRU Reports 50 and 62 [15], [16]:

- **Gross Tumor Volume (GTV):** The visible or palpable extent of the tumor as identified on imaging (typically CT). The GTV represents the region of highest tumor cell density and is the primary target for this thesis.
- **Clinical Target Volume (CTV):** The CTV plus a margin accounting for microscopic disease extension that is not visible on imaging **giraud2002evaluation**.
- **Planning Target Volume (PTV):** The PTV plus an additional margin accounting for geometric uncertainties such as patient setup errors and organ motion during treatment.

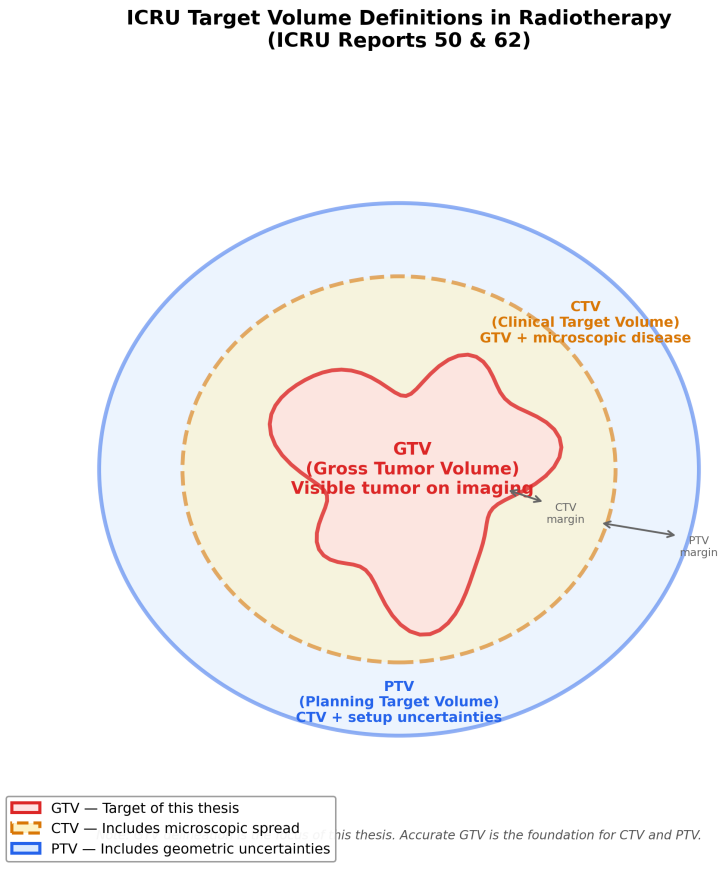

Figure 1.1: ICRU target volume definitions. The GTV represents the visible tumor extent and is the segmentation target in this thesis. The CTV and PTV include additional margins for microscopic disease and geometric uncertainties, respectively (ICRU Reports 50 and 62).

Accurate delineation of the GTV is the foundation upon which all subsequent volumes (CTV, PTV) are built. Errors in GTV delineation propagate through the entire planning chain, potentially leading to geographic miss of the tumor or unnecessary irradiation of normal tissue.

1.2.3 The Challenge of Manual GTV Delineation

Manual GTV delineation on CT images is a labor-intensive task that typically requires 30 to 90 minutes per patient, depending on the tumor complexity and the experience of the clinician [3]. Furthermore, the process is inherently subjective: studies have shown significant inter-observer variability, with different clinicians producing substantially different contours for the same patient **giraud2002evaluation**.

This variability has direct clinical consequences. Differences in GTV delineation can lead to differences in the prescribed dose distribution, potentially affecting treatment outcomes. A recent multi-institutional study reported Dice coefficients as low as 0.50–0.60 between contours drawn by different experts for the same NSCLC patients [4], highlighting the magnitude of this problem.

Automated or semi-automated delineation tools based on deep learning offer a promising solution to both challenges: they can reduce contouring time from tens of minutes to seconds, and they provide consistent, reproducible results that are not subject to inter-observer variability [7].

1.3 Medical Imaging Fundamentals

1.3.1 Computed Tomography (CT)

Computed Tomography is the primary imaging modality used for radiotherapy planning. A CT scanner produces cross-sectional images of the body by rotating an X-ray source and detector array around the patient and reconstructing the attenuation data into a three-dimensional volume [17].

The resulting image represents a volumetric map of X-ray attenuation coefficients, expressed in Hounsfield Units (HU). The HU scale is defined such that water has a value of 0 HU and air has a value of -1000 HU [17]:

$$\mathrm{HU} = 1000 \times \frac{\mu_{\mathrm{tissue}} - \mu_{\mathrm{water}}}{\mu_{\mathrm{water}}}, \tag{1.1}$$

where μ denotes the linear attenuation coefficient. Table 1.1 lists the approximate HU values for tissues relevant to thoracic CT imaging.

A typical thoracic CT scan consists of 100–200 axial slices, each of 512×512 pixels, with an in-plane resolution of 0.5–1.0 mm and a slice thickness of 1.0–5.0 mm [17].

Table 1.1: Typical Hounsfield Unit values for thoracic tissues.

Tissue / Material	**HU Range**
Air	-1000
Lung parenchyma	-900 to -500
Fat	-120 to -60
Water	0
Soft tissue	$+20$ to $+80$
Tumor (NSCLC)	$+20$ to $+200$
Bone (cortical)	$+300$ to $+3000$

1.3.2 DICOM Standard

Medical images are stored and transmitted using the Digital Imaging and Communications in Medicine (DICOM) standard, which defines both a file format and a communication protocol [18]. A DICOM file contains not only the pixel data but also extensive metadata, including:

- Patient information (name, ID, age, sex)
- Acquisition parameters (kVp, mAs, slice thickness)
- Geometric information (pixel spacing, image position, image orientation)
- Modality-specific data (rescale slope and intercept for conversion to HU)

In the context of radiotherapy, several specialized DICOM objects are used:

- **CT Image:** The volumetric scan used for planning.
- **RTSTRUCT (RT Structure Set):** Contains the contours of delineated structures (GTV, OARs) as a set of polygonal outlines on each axial slice.
- **RTDOSE:** The three-dimensional dose distribution computed by the Treatment Planning System.
- **RTPLAN:** The treatment plan parameters (beam angles, energies, monitor units).

This thesis makes use of CT and RTSTRUCT objects. The RTSTRUCT format encodes tumor contours as ordered sequences of (x, y, z) coordinates on each axial slice. These contours are converted to binary volumetric masks during preprocessing (Section 2.3). The NSCLC-Radiomics dataset used in this work is hosted on The Cancer Imaging Archive (TCIA) [10] and was originally published as part of the radiomics study by Aerts et al. [9].

1.4 Deep Learning for Medical Image Segmentation

1.4.1 From Machine Learning to Deep Learning

Traditional machine learning approaches to image segmentation relied on hand-crafted features (intensity histograms, texture descriptors, shape models) combined with classifiers such as random forests or support vector machines. While effective for certain tasks, these methods required extensive domain-specific feature engineering and often failed to generalize across different imaging conditions.

Deep learning, and in particular Convolutional Neural Networks (CNNs), marked a paradigm shift by learning hierarchical feature representations directly from raw image data [19]. Since the seminal work of Krizhevsky et al. [20] on large-scale image classification, deep learning has progressively become the dominant approach in medical image analysis [19].

1.4.2 Convolutional Neural Networks (CNN)

A CNN is a neural network architecture specifically designed to process data with a grid-like topology, such as images [21]. The key components of a CNN are:

Convolutional Layer

The convolutional layer applies a set of learnable filters (kernels) to the input, producing feature maps that capture local patterns [21]. For a 3D input volume $\mathbf{X}$ and a kernel $\mathbf{W}$ of size $k \times k \times k$, the convolution operation at position (i, j, l) is:

$$(\mathbf{X} * \mathbf{W})(i, j, l) = \sum_{a} \sum_{b} \sum_{c} \mathbf{X}(i + a, j + b, l + c) \cdot \mathbf{W}(a, b, c) + \text{bias}, \tag{1.2}$$

where the summation runs over the kernel dimensions. The use of shared weights across spatial positions significantly reduces the number of parameters compared to fully connected layers and introduces translation equivariance [21].

Activation Functions

Non-linear activation functions are applied after each convolution to enable the network to learn complex mappings. The most commonly used activation in modern CNNs is the Rectified Linear Unit (ReLU):

$$\text{ReLU}(x) = \max(0, x). \tag{1.3}$$

ReLU is computationally efficient and mitigates the vanishing gradient problem that affects deeper networks [21].

Pooling Layer

Pooling layers reduce the spatial dimensions of the feature maps, thereby reducing the computational cost and providing a degree of spatial invariance. Max-pooling selects the maximum value within a local window:

$$\text{MaxPool}(\mathbf{X})(i,j) = \max_{(a,b)\in\mathcal{R}} \mathbf{X}(i+a, j+b), \tag{1.4}$$

where $\mathcal{R}$ denotes the pooling region (typically 2×2 or $2 \times 2 \times 2$ in 3D).

Batch Normalization

Batch normalization normalizes the activations within each mini-batch to have zero mean and unit variance, followed by a learned affine transformation [22]:

$$\hat{x}_i = \gamma \cdot \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta, \tag{1.5}$$

where μ_B and σ_B^2 are the mini-batch mean and variance, and γ, β are learnable scale and shift parameters. Batch normalization accelerates convergence and acts as a regularizer [22].

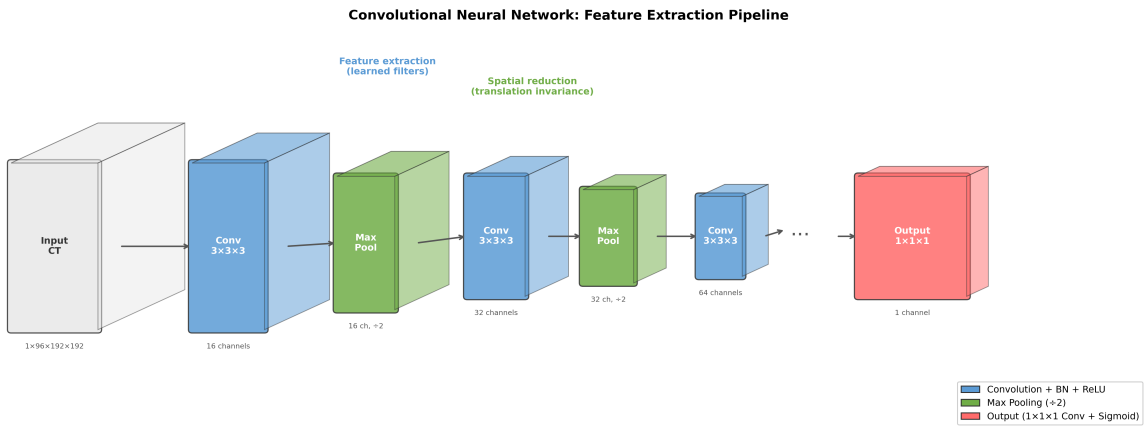

Figure 1.2: Schematic illustration of a CNN feature extraction pipeline. Successive convolutional layers extract increasingly abstract features, while pooling layers reduce spatial dimensions.

1.4.3 Image Segmentation

Image segmentation is the task of assigning a class label to each pixel (or voxel, in 3D) of an image. In medical imaging, segmentation is used to delineate anatomical

structures or pathological regions of interest **liu2021review**.

The two main deep learning paradigms for segmentation are:

- **Semantic segmentation:** Each voxel is assigned a class label from a predefined set (e.g., background, tumor, lung, heart). The present work performs *binary* semantic segmentation (GTV vs. background).
- **Instance segmentation:** Individual instances of the same class are distinguished (e.g., separate tumor lesions). This is not addressed in this thesis.

1.4.4 The U-Net Architecture

Original 2D U-Net

The U-Net, introduced by Ronneberger et al. in 2015 [5], was specifically designed for biomedical image segmentation. Its architecture consists of two symmetric paths:

- An **encoder** (contracting path) that progressively reduces spatial resolution while increasing the number of feature channels, capturing contextual information at multiple scales.
- A **decoder** (expanding path) that progressively restores the spatial resolution using upsampling operations, producing a segmentation map at the original input resolution.

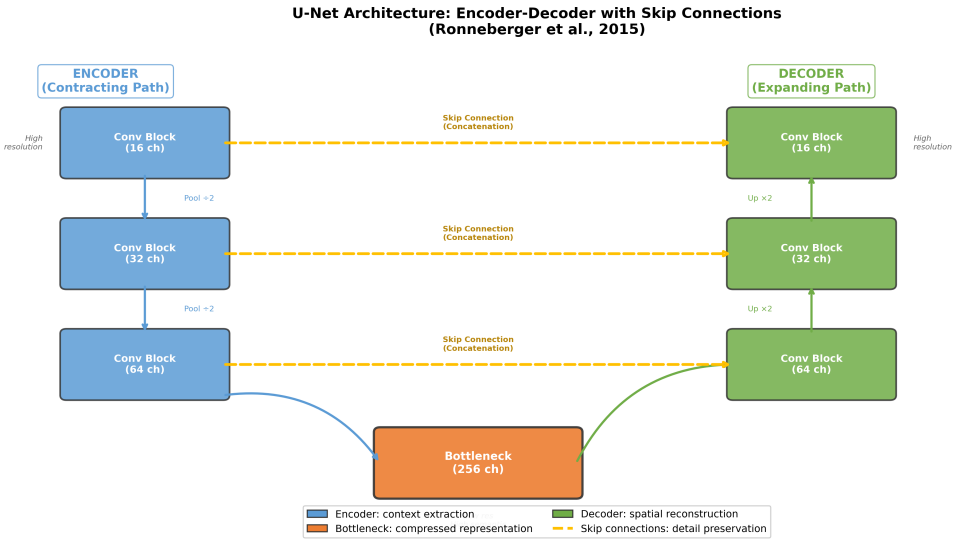

Figure 1.3: Conceptual illustration of the U-Net architecture showing the encoder (contracting) path, bottleneck, decoder (expanding) path, and skip connections that preserve spatial details [5].

The defining characteristic of U-Net is the use of **skip connections** that directly connect encoder layers to their corresponding decoder layers at each resolution level.

These connections allow the decoder to combine high-level semantic information from the deep layers with high-resolution spatial details from the shallow layers, enabling precise boundary localization [5].

Since its introduction, U-Net has become the de facto standard architecture for medical image segmentation, achieving state-of-the-art results across a wide range of tasks including cell segmentation, organ delineation, and tumor detection **liu2021review**, [5].

3D U-Net

The extension of U-Net to three dimensions was proposed by Çiçek et al. in 2016 [6]. The 3D U-Net replaces all 2D operations (convolutions, pooling, upsampling) with their 3D counterparts, enabling the network to exploit spatial context along all three axes. This is particularly important for volumetric medical data such as CT scans, where adjacent axial slices contain correlated information that would be lost by a slice-by-slice 2D approach [6].

The main challenge of 3D U-Net is its substantially higher computational and memory requirements compared to 2D U-Net. A single 3D convolution with k^3 kernel elements requires k times more computation than its 2D counterpart (k^2 elements). This is typically addressed by using smaller input volumes (cropping) and fewer feature channels.

Variants and Extensions

Numerous variants of U-Net have been proposed to improve its performance:

- **Attention U-Net:** Incorporates attention gates that learn to focus on relevant regions, suppressing irrelevant features [23].
- **Residual U-Net:** Replaces standard convolutional blocks with residual blocks, facilitating the training of deeper networks [24].
- **nnU-Net** [7]: A self-configuring framework that automatically adapts preprocessing, architecture, and training parameters to the dataset at hand. It has established new benchmarks on numerous medical segmentation challenges.
- **Transformer-based approaches:** Recent architectures such as Swin UNETR combine the U-Net topology with vision transformers, capturing long-range dependencies that CNNs may miss [25].

This thesis employs a standard 3D U-Net architecture, as detailed in Section 2.4, prioritizing simplicity and computational efficiency given the constraints of the available computing resources.

1.5 State of the Art in Automatic GTV Segmentation

The automatic segmentation of Gross Tumor Volume in lung cancer has been an active area of research for over a decade. This section reviews the most relevant approaches, organized chronologically from early methods to recent deep learning-based solutions.

1.5.1 Early Approaches

Prior to the widespread adoption of deep learning, several approaches were explored for semi-automatic or automatic tumor delineation:

- **Atlas-based methods:** These approaches register a new patient's CT to a set of pre-segmented atlas images and propagate the known segmentations via the computed deformation field. While effective for organs with relatively stable anatomy (e.g., brain structures), atlas-based methods perform poorly for tumors due to the high variability in tumor location, size, and morphology across patients.
- **Statistical shape models:** These methods capture the typical shape variability of a structure from a training set and constrain the segmentation to plausible shapes. Their applicability to tumors is limited by the irregular and patient-specific nature of tumor geometry.
- **Threshold and region-growing methods:** Simple intensity-based approaches that exploit the contrast between tumor and surrounding tissue. These methods are sensitive to imaging noise and fail when the tumor has similar intensity to adjacent structures (e.g., mediastinal tumors adjacent to soft tissue).

The limited success of these classical approaches motivated the transition to learning-based methods capable of capturing complex, non-linear relationships between image features and segmentation labels.

1.5.2 Deep Learning Approaches

The application of deep learning to lung tumor segmentation has yielded progressively improving results over the past decade. Table 1.2 summarizes the key studies.

Table 1.2: Summary of representative deep learning-based studies for lung tumor / GTV segmentation. Reported DSC values correspond to each study's primary test result as stated in the cited publication.

Study	**Architecture**	**Dataset (# cases)**	**DSC**
Jiang et al. (2019) [26]	incremental-MRRN	TCIA NSCLC (377)	0.74 ± 0.13
Kamal et al. (2019) [27]	Recurrent 3D-DenseUNet	NSCLC (MSD)	0.72
Ferrante et al. (2022) [28]	nnU-Net (2D+3D)	Multi-cohort (899)	0.78 ± 0.12
Atiya & Ramesh (2024) [29]	3D UNet-DeepLab (CT+PET)	NSCLC-Radiomics	0.75
Kulkarni et al. (2024) [8]	nnU-Net	NSCLC-Radiomics (402)	0.65–0.75
This work (2025)	**3D U-Net**	**NSCLC-Radiomics (45)**	**0.547**
This work — large tumors	**3D U-Net**	**NSCLC-Radiomics (45)**	**0.750**

Jiang et al. (2019)

Jiang et al. [26] introduced the Multiple Resolution Residually Connected Network (MRRN), which combines features across multiple image resolutions through residual connections. Training on 377 tumors from the TCIA dataset, their best-performing incremental-MRRN achieved a mean DSC of 0.74 ± 0.13 on TCIA data, outperforming standard U-Net and SegNet baselines. Their work is particularly relevant as it uses the same public data source (TCIA) as the present study, and it explicitly reported lower performance on very small tumors—a finding directly consistent with the size-stratified results presented in this thesis (Section 3.4).

Kamal et al. (2020)

Kamal et al. [27] proposed a Recurrent 3D-DenseUNet that incorporates convolutional LSTM units to exploit inter-slice continuity in volumetric CT data. The network achieved an average DSC of approximately 0.72, and the authors conducted an extensive ablation study of different loss functions, highlighting the importance of loss-function design for the class-imbalanced tumor segmentation task.

Ferrante et al. (2022)

Ferrante et al. [28] applied the nnU-Net framework to a large multi-cohort dataset of 899 patients drawn from two proprietary and one public dataset.

Atiya & Ramesh (2024)

Atiya & Ramesh. [29] proposed a multi-modal fusion approach combining CT and PET images within a 3D UNet-DeepLab architecture, evaluated on the NSCLC-Radiomics dataset. Their model achieved a DSC of 0.7502. It should be noted, however, that the incorporation of PET data provides additional metabolic information that is not available in the present CT-only study, which partly explains the higher performance.

Kulkarni et al. (2024)

Kulkarni et al. [8] conducted a particularly relevant study using the nnU-Net framework on the NSCLC-Radiomics dataset. Their work focused on the transferability of segmentation models across institutions, showing that a model trained on TCIA data achieved DSC values of 0.65–0.75, and that performance could be improved through transfer learning with a relatively small number of local cases. This finding is directly relevant to the present work, as it suggests that the performance gap between our model and the state of the art could be substantially narrowed by increasing the training set size.

1.5.3 Key Observations from the Literature

Several consistent findings emerge from the reviewed studies:

1. **3D approaches outperform 2D:** All recent state-of-the-art results employ volumetric (3D) architectures, confirming the importance of axial context for tumor segmentation.

2. **Training data volume is critical:** Studies using fewer than 100 patients consistently report lower DSC values than those using 200 or more patients.

3. **Small tumors remain challenging:** Multiple studies report significantly lower performance on small tumors compared to large ones, a finding confirmed in this thesis (Section 3.4).

4. **The U-Net family dominates:** Despite the introduction of transformers and other architectures, U-Net variants remain the most widely used and competitive architectures for medical image segmentation.

5. **Class imbalance handling is essential:** Studies that do not explicitly address the foreground-background imbalance consistently report convergence to trivial solutions, as observed in the initial training experiment described in Section 2.5.

1.5.4 Positioning of This Work

This thesis contributes to the existing body of work in several ways:

- It demonstrates that a standard 3D U-Net architecture can achieve competitive performance on large tumors (DSC = 0.750) even when trained on a limited dataset (45 patients) with constrained computational resources (Google Colab Free).
- It provides a detailed analysis of the relationship between tumor volume and segmentation accuracy, a factor that is often reported only in aggregate in the literature.
- It documents the failure of Test Time Augmentation with spatial flipping in thoracic CT segmentation, contributing a negative result that may guide future methodological choices.
- It presents a fully reproducible pipeline, from data acquisition to evaluation, that can serve as a baseline for future improvements.

Chapter 2

Materials and Methods

This chapter presents the methodology adopted for the automatic segmentation of Gross Tumor Volume (GTV) in Non-Small Cell Lung Cancer patients using deep learning. The chapter is organized as follows: Section 2.1 provides an overview of the proposed pipeline. Section 2.2 describes the dataset and patient selection criteria. Section 2.3 details the preprocessing steps. Section 2.4 presents the network architecture. Section 2.5 describes the training configuration, and Section 2.6 outlines the evaluation methodology.

2.1 Overview of the Proposed Pipeline

The proposed pipeline consists of four main stages: (1) data acquisition and selection from the publicly available NSCLC-Radiomics dataset; (2) preprocessing of CT volumes and extraction of GTV masks from RTSTRUCT files; (3) training of a three-dimensional U-Net (3D U-Net) network for binary segmentation; and (4) quantitative and qualitative evaluation on an independent test set.

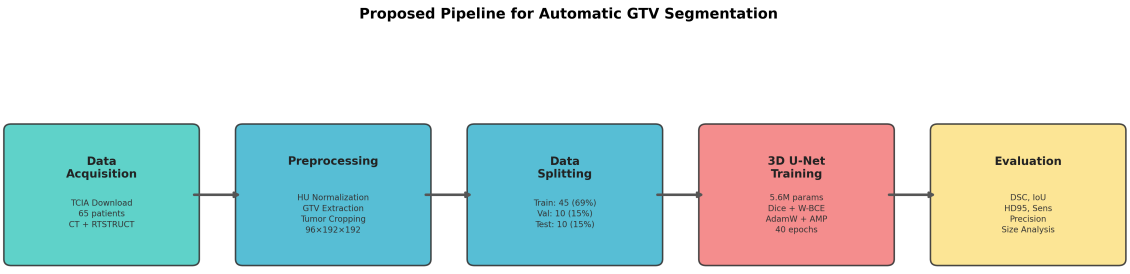

Figure 2.1: Overview of the proposed pipeline for automatic GTV segmentation, from data acquisition to evaluation.

The pipeline was designed with two primary considerations in mind: clinical rel-

evance and computational efficiency. The choice of 3D U-Net was motivated by its well-established performance in volumetric medical image segmentation, while the use of cloud-based computing resources (Google Colab) ensured reproducibility and accessibility of the proposed method.

2.2 Dataset

2.2.1 NSCLC-Radiomics Collection

The NSCLC-Radiomics dataset [9] was used in this study. This publicly available collection, hosted on The Cancer Imaging Archive (TCIA) [10], comprises pre-treatment computed tomography (CT) scans of 422 patients with Non-Small Cell Lung Cancer (NSCLC). For each patient, a manual delineation of the primary tumor (GTV-1) and selected anatomical structures was performed by an experienced radiation oncologist and stored in DICOM Radiation Therapy Structure Set (RTSTRUCT) format.

2.2.2 Patient Selection

From the original cohort of 422 patients, a subset was selected based on the availability of paired CT and RTSTRUCT data. The data acquisition process, described in Section 2.3, resulted in 67 patient cases being downloaded. Among these:

- 67 cases had a complete CT series,
- 66 cases had an associated RTSTRUCT file,
- 67 cases had a Segmentation (SEG) file.

After verifying the integrity of CT-RTSTRUCT pairs and successful extraction of the GTV-1 contour, the final cohort comprised **65 patients** (one case was excluded due to a reference frame mismatch between the CT series and the RTSTRUCT file).

2.2.3 Tumor Volume Characteristics

The final cohort exhibited considerable variability in tumor size, reflecting the heterogeneity encountered in clinical practice. The GTV-1 volumes ranged from approximately $1.0\,\mathrm{cm}^3$ to over $670\,\mathrm{cm}^3$, with a mean of approximately $104\,\mathrm{cm}^3$. This wide distribution provides a challenging benchmark for the segmentation model and allows performance stratification by tumor size, as discussed in Chapter 3.

2.3 Data Acquisition and Preprocessing

2.3.1 Data Acquisition from TCIA

The dataset was downloaded from TCIA using the official NBIA manifest file (version 4, October 2020), which contains 1,265 DICOM series (`SeriesInstanceUIDs`) corresponding to CT, RTSTRUCT, and Segmentation data. A subset of 200 series was selected for this study, which corresponded to 67 patient cases after grouping by `PatientID`.

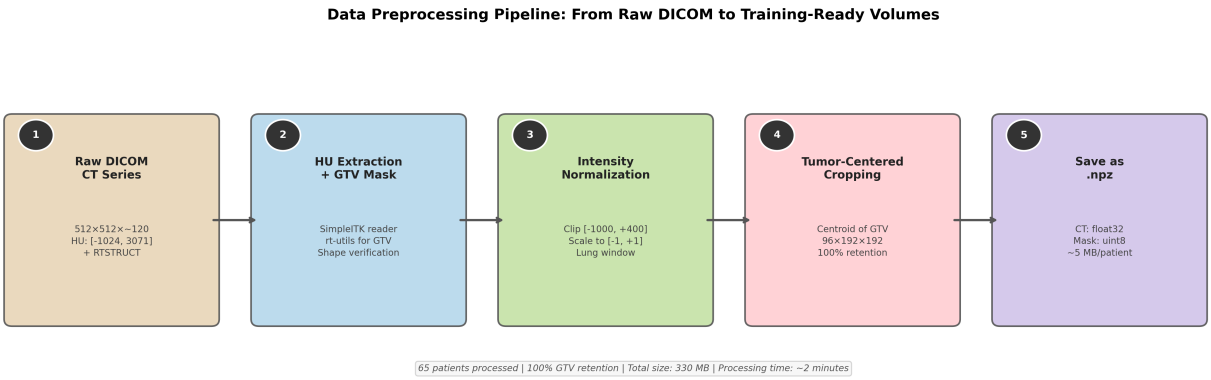

Figure 2.2: Data preprocessing pipeline showing the five main steps: DICOM extraction, HU normalization, tumor-centered cropping, and storage in compressed NumPy format.

The download was performed programmatically using the `tcia_utils` Python library, which interfaces with the TCIA REST API. The complete acquisition process took approximately 10 minutes and resulted in a total of 5.55 GB of raw DICOM data comprising 8,147 individual files.

2.3.2 CT Volume Extraction

For each patient, the CT volume was reconstructed from the corresponding DICOM series using the `SimpleITK` library [30]. SimpleITK was selected over alternative readers (such as `pydicom`) because it automatically applies the rescale slope and intercept transformations specified in the DICOM headers, ensuring that the resulting voxel values are expressed in correct Hounsfield Units (HU).

The original CT volumes exhibited typical dimensions of approximately 512×512 pixels in-plane, with the number of axial slices varying between 100 and 140 per patient depending on the field of view. The voxel spacing was typically $0.977 \times 0.977 \times 3.0$ mm (in-plane $\times$ slice thickness).

2.3.3 GTV Mask Extraction from RTSTRUCT

The GTV-1 region of interest (ROI) was extracted from each patient's RTSTRUCT file using the `rt-utils` library. This library reconstructs a binary three-dimensional mask from the polygonal contours stored in the RTSTRUCT, ensuring spatial alignment with the corresponding CT volume.

The resulting GTV mask has the same shape and voxel spacing as the CT volume, with voxel values of 1 inside the tumor region and 0 elsewhere. The integrity of the CT-mask alignment was verified for each patient by visual inspection of representative axial slices, as illustrated in Figure **??** (to be added).

2.3.4 Intensity Normalization

To prepare the CT volumes for input to the neural network, a two-step intensity normalization was applied:

1. **Hounsfield Unit clipping:** CT values were clipped to the range $[-1000, +400]$ HU. This range was selected to preserve the relevant anatomical structures (lung parenchyma at approximately -800 HU, soft tissue near 0 HU, and tumor tissue typically between 0 and +200 HU) while excluding extreme values from metallic artifacts and air outside the body.

2. **Linear normalization:** The clipped values were then linearly normalized to the range $[-1, +1]$ using the transformation:

$$\mathrm{HU}_{\mathrm{norm}} = 2 \cdot \frac{\mathrm{HU} - \mathrm{HU}_{\mathrm{min}}}{\mathrm{HU}_{\mathrm{max}} - \mathrm{HU}_{\mathrm{min}}} - 1, \tag{2.1}$$

   where $\mathrm{HU}_{\mathrm{min}} = -1000$ and $\mathrm{HU}_{\mathrm{max}} = +400$. This normalization is suitable for networks employing tanh-like activation behavior in their early layers.

2.3.5 Tumor-Centered Cropping

Given the computational constraints of training a fully volumetric 3D network, full-resolution CT volumes ($512 \times 512 \times \sim 120$) could not be processed directly. A center-of-mass cropping strategy was therefore employed:

1. For each patient, the centroid of the GTV-1 mask was computed by averaging the coordinates of all positive voxels.

2. A volume of size $96 \times 192 \times 192$ voxels (depth $\times$ height $\times$ width) was extracted centered on this centroid.

3. When the requested volume extended beyond the original boundaries, zero-padding was applied (with a value of -1000 HU for the CT, corresponding to air, and 0 for the mask).

This cropping strategy was validated by computing the proportion of tumor voxels retained after cropping. For all 65 patients in the final cohort, the retention was 100 %, confirming that no portion of the GTV was lost during this step.

2.3.6 Final Dataset Statistics

After preprocessing, each patient was represented by a pair of NumPy arrays of identical shape $(96, 192, 192)$: a normalized CT volume (`float32`, values in $[-1, +1]$) and a binary GTV mask (`uint8`, values in $\{0, 1\}$). The processed data was stored in compressed `.npz` format, resulting in a total storage requirement of approximately 330 MB for all 65 patients.

Table 2.1 summarizes the key characteristics of the final dataset.

Table 2.1: Summary statistics of the preprocessed dataset.

Characteristic	**Value**
Number of patients	65
Volume dimensions (after cropping)	$96 \times 192 \times 192$
Voxel spacing	$3.0 \times 0.977 \times 0.977$ mm
CT value range (after normalization)	$[-1, +1]$
GTV volume range	349 – 235,919 voxels
GTV volume mean (std)	36,340 ($\sim$ 41,000) voxels
GTV volume median	29,537 voxels
Storage format	Compressed NumPy (`.npz`)
Total dataset size	329.9 MB

Note: voxel counts can be converted to physical volume using the voxel size of $3.0 \times 0.977 \times 0.977 \approx 2.86\,\text{mm}^3$; for example, the mean of 36,340 voxels corresponds to approximately $104\,\text{cm}^3$.

2.4 Network Architecture

2.4.1 Design Rationale

The 3D U-Net architecture [6], an extension of the original 2D U-Net [5], was selected for this work due to its proven effectiveness in volumetric medical image segmentation. Three characteristics make it particularly suited to the present task:

- **Three-dimensional convolutions** that exploit the inherent volumetric nature of CT data, capturing spatial context along the axial direction in addition to the in-plane information.
- **Skip connections** between encoder and decoder paths, which preserve high-resolution spatial details that would otherwise be lost through successive down-sampling operations.
- **Symmetric encoder-decoder structure** that produces output segmentations at the same resolution as the input, enabling voxel-wise classification.

2.4.2 Overall Architecture

The proposed network follows a five-level U-shaped topology, comprising four encoder blocks, a bottleneck, and four decoder blocks. Figure 2.3 illustrates the overall structure. The network accepts a single-channel input of size $96 \times 192 \times 192$ (corresponding to a normalized CT volume) and produces a single-channel output of identical size, representing the logits of the GTV segmentation map. A sigmoid activation is applied at inference time to obtain probabilities in $[0, 1]$.

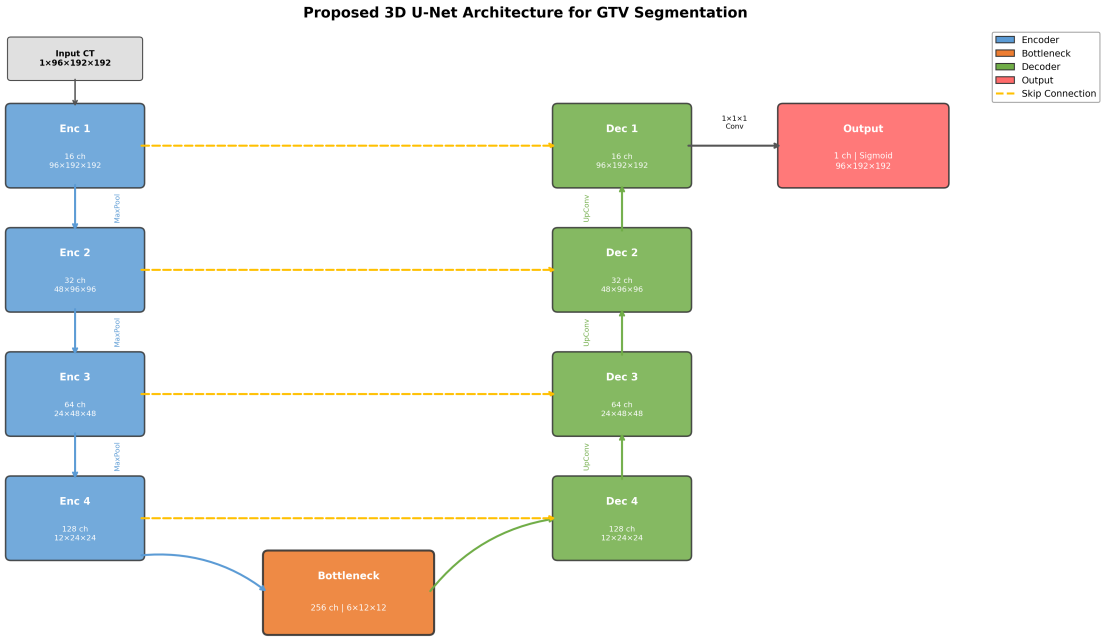

Figure 2.3: Detailed architecture of the proposed 3D U-Net. The encoder (blue) progressively reduces spatial dimensions while increasing channels (16→256). The decoder (green) restores resolution via transposed convolutions. Skip connections (yellow dashed) concatenate encoder features with decoder features at each level.

2.4.3 Building Blocks

Convolutional Block

The fundamental building block of the network, denoted `ConvBlock3D`, consists of two consecutive sequences of: a $3 \times 3 \times 3$ three-dimensional convolution with padding of 1 (to preserve spatial dimensions), followed by batch normalization and a Rectified Linear Unit (ReLU) activation. Formally, for an input tensor $\mathbf{x}$:

$$\text{ConvBlock3D}(\mathbf{x}) = \text{ReLU}(\text{BN}(\text{Conv}_2(\text{ReLU}(\text{BN}(\text{Conv}_1(\mathbf{x})))))). \tag{2.2}$$

Batch normalization stabilizes training by normalizing the activations across each mini-batch, while the ReLU activation introduces the non-linearity required to learn complex representations. The convolutions were configured without bias terms, as their effect is absorbed by the subsequent batch normalization.

Downsampling

Downsampling between encoder levels is performed by max-pooling with a kernel size of 2 and stride of 2, which halves the spatial dimensions along each axis. Max-pooling was preferred over strided convolutions because of its lower parameter count and its ability to preserve the most salient features within each local region.

Upsampling

Upsampling in the decoder path is performed by transposed convolutions with a kernel size of 2 and stride of 2. Unlike simple interpolation methods (e.g., trilinear upsampling), transposed convolutions are learnable, allowing the network to optimize the upsampling operation for the specific task.

2.4.4 Encoder Path

The encoder progressively reduces the spatial resolution while increasing the number of feature channels, allowing the network to capture increasingly abstract representations. With a base channel count $C_0 = 16$, the channel progression is $16 \to 32 \to 64 \to 128$ across the four encoder levels, with the bottleneck reaching 256 channels.

Each encoder level applies a `ConvBlock3D` followed by max-pooling, except for the deepest level (bottleneck), where pooling is omitted.

2.4.5 Decoder Path with Skip Connections

The decoder mirrors the encoder structure, with each level performing the following operations:

1. **Upsample** the feature map from the previous level using transposed convolution, doubling the spatial dimensions and halving the channels.
2. **Concatenate** the upsampled feature map with the corresponding encoder feature map of the same resolution (skip connection).
3. **Apply** a `ConvBlock3D` to the concatenated tensor.

The skip connections are essential for recovering fine spatial details that would be lost in the encoder. Without them, the network would be unable to produce accurate segmentation boundaries.

2.4.6 Output Layer

The final layer is a $1\times1\times1$ convolution that reduces the feature dimension from $C_0 = 16$ to a single output channel. The output represents *logits* (pre-activation values), which are converted to probabilities by a sigmoid activation:

$$p(\mathbf{x}) = \sigma(\text{logits}(\mathbf{x})) = \frac{1}{1 + e^{-\text{logits}(\mathbf{x}))}}. \tag{2.3}$$

A binary segmentation map is then obtained by thresholding p at 0.5.

2.4.7 Architectural Details

Table 2.2 summarizes the layer-by-layer specifications of the network. The total number of trainable parameters is approximately 5.65×10^6, corresponding to a model size of $\sim$22.6 MB when stored in single-precision floating-point format (`float32`).

2.5 Training Configuration

2.5.1 Dataset Splitting

The 65 preprocessed patient cases were divided into three disjoint subsets using a stratified random splitting strategy with a fixed random seed (42) to ensure reproducibility:

- **Training set:** 45 patients (69 %) used for optimizing the network parameters.
- **Validation set:** 10 patients (15 %) used for hyperparameter selection, learning rate scheduling, and early stopping.
- **Test set:** 10 patients (15 %) held out entirely from the training process and used only for the final evaluation reported in Chapter 3.

The splits were performed at the patient level rather than at the slice level, ensuring that no information from any test patient could leak into the training process.

Table 2.2: Layer-by-layer specification of the proposed 3D U-Net.

Stage	**Operation**	**Output Shape**	**Channels**
Input	–	$96 \times 192 \times 192$	1
Encoder 1	ConvBlock3D	$96 \times 192 \times 192$	16
	MaxPool	$48 \times 96 \times 96$	16
Encoder 2	ConvBlock3D	$48 \times 96 \times 96$	32
	MaxPool	$24 \times 48 \times 48$	32
Encoder 3	ConvBlock3D	$24 \times 48 \times 48$	64
	MaxPool	$12 \times 24 \times 24$	64
Encoder 4	ConvBlock3D	$12 \times 24 \times 24$	128
	MaxPool	$6 \times 12 \times 12$	128
Bottleneck	ConvBlock3D	$6 \times 12 \times 12$	256
Decoder 4	TransposeConv + Concat	$12 \times 24 \times 24$	256
	ConvBlock3D	$12 \times 24 \times 24$	128
Decoder 3	TransposeConv + Concat	$24 \times 48 \times 48$	128
	ConvBlock3D	$24 \times 48 \times 48$	64
Decoder 2	TransposeConv + Concat	$48 \times 96 \times 96$	64
	ConvBlock3D	$48 \times 96 \times 96$	32
Decoder 1	TransposeConv + Concat	$96 \times 192 \times 192$	32
	ConvBlock3D	$96 \times 192 \times 192$	16
Output	Conv $(1 \times 1 \times 1)$	$96 \times 192 \times 192$	1

2.5.2 Data Augmentation

To improve generalization given the limited training set size, a lightweight on-the-fly data augmentation strategy was applied during training:

- **Random horizontal flip:** The volume was flipped along the lateral (left-right) axis with probability 0.5. This augmentation exploits the approximate bilateral symmetry of the thoracic anatomy.
- **Intensity jitter:** The normalized intensity values were multiplied by a random factor drawn uniformly from $[0.9, 1.1]$ with probability 0.5, then clipped back to $[-1, +1]$. This simulates minor variations in CT scanner calibration.

Stronger augmentations (rotations, elastic deformations) were avoided to preserve the anatomically meaningful orientation of the CT volumes.

2.5.3 Loss Function

The choice of loss function is critical in medical image segmentation, particularly in the presence of severe class imbalance. In the present dataset, the GTV typically occupies only 0.1 to 1.0% of the voxels within the cropped volume, resulting in a foreground-to-background ratio of approximately 1:250 on average. Without appropriate handling,

a naive loss function would lead the network to converge to the trivial solution of predicting all voxels as background.

To address this challenge, a combined loss function was employed, integrating two complementary components:

Dice Loss

The Dice Loss is defined as the complement of the soft Dice coefficient [31]:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_i p_i\, g_i + \epsilon}{\sum_i p_i + \sum_i g_i + \epsilon}, \tag{2.4}$$

where $p_i \in [0, 1]$ is the predicted probability for voxel i, $g_i \in \{0, 1\}$ is the ground truth label, and $\epsilon = 1$ is a smoothing term that prevents division by zero. The Dice Loss is inherently robust to class imbalance, as it considers the relative overlap rather than absolute voxel-wise errors.

Weighted Binary Cross-Entropy Loss

The standard Binary Cross-Entropy (BCE) Loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\Big[w \cdot g_i \log(p_i) + (1 - g_i)\log(1 - p_i)\Big], \tag{2.5}$$

where N is the total number of voxels, and w is a positive class weight that amplifies the contribution of foreground voxels. Empirically, $w = 100$ was selected to compensate for the class imbalance ratio.

Combined Loss

The total training loss is a weighted sum of the two components:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{Dice}} + (1 - \alpha) \cdot \mathcal{L}_{\text{BCE}}, \tag{2.6}$$

with $\alpha = 0.7$ giving slightly more weight to the Dice component, which is more directly aligned with the segmentation metric of interest.

Justification of the loss design. An initial experiment was conducted using equal weights ($\alpha = 0.5$) and a standard BCE without class weighting ($w = 1$). After eight epochs of training, the network had converged to a near-zero Dice score on both the training and validation sets, predicting all voxels as background. This confirmed the necessity of explicit class-imbalance handling and motivated the adoption of the weighted formulation. The chosen configuration ($\alpha = 0.7$, $w = 100$) yielded a productive learning trajectory within the first three epochs of subsequent training.

2.5.4 Optimizer and Learning Rate Scheduling

The network was optimized using the AdamW algorithm [32] with the following hyperparameters:

- Initial learning rate: 10^{-3}
- Weight decay (L2 regularization): 10^{-5}
- $\beta_1 = 0.9$, $\beta_2 = 0.999$ (default values)

A `ReduceLROnPlateau` scheduler was employed to adapt the learning rate based on validation performance. The scheduler reduces the learning rate by a factor of 0.5 when the validation Dice score has not improved for five consecutive epochs, with a minimum learning rate of 10^{-6}.

2.5.5 Training Stability Techniques

Two additional techniques were employed to ensure stable training:

- **Mixed precision training** [33] using PyTorch's [34] Automatic Mixed Precision (AMP) framework. This combines `float16` computations for forward and backward passes with `float32` parameter updates, reducing memory consumption by approximately 50 % and accelerating training on modern GPUs without sacrificing numerical stability.
- **Gradient clipping** with a maximum norm of 1.0, applied before each optimizer step to prevent occasional gradient explosions caused by the high `pos_weight` value.

2.5.6 Early Stopping and Model Selection

Training was configured for a maximum of 50 epochs. An early stopping criterion was applied: training was halted if the validation Dice score failed to improve for 10 consecutive epochs. The model checkpoint corresponding to the highest validation Dice score across all epochs (the "best model") was retained for the final evaluation.

2.5.7 Hyperparameter Summary

Table 2.3 summarizes all the hyperparameters used during training.

Table 2.3: Summary of training hyperparameters.

Hyperparameter	**Value**
Batch size	2
Number of epochs (max)	50
Early stopping patience	10 epochs
Initial learning rate	10^{-3}
Weight decay	10^{-5}
LR scheduler	ReduceLROnPlateau
reduction factor	0.5
patience	5 epochs
minimum LR	10^{-6}
Loss function	Combined Dice + Weighted BCE
Dice weight (α)	0.7
BCE positive class weight	100
Smoothing term (ϵ)	1.0
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Gradient clipping	max norm = 1.0
Mixed precision	Enabled (AMP)
Random seed	42

2.5.8 Implementation Environment

All experiments were conducted in Python (version 3.10) using the PyTorch framework [34] (version 2.10) on Google Colaboratory. The training was performed on a single NVIDIA Tesla T4 GPU with 15.6 GB of VRAM. With the configuration described above, one complete training epoch (45 patients, 23 mini-batches) took approximately 15–17 seconds, and the full training process (40 epochs before early stopping) was completed in approximately 10.7 minutes.

2.6 Evaluation Methodology

2.6.1 Inference Pipeline

At inference time, each test volume undergoes the same preprocessing pipeline described in Section 2.3: HU clipping, linear normalization to $[-1, +1]$, and tumor-centered cropping to $96 \times 192 \times 192$ voxels. The preprocessed volume is then passed through the trained network, producing a tensor of logits of identical shape.

A sigmoid activation is applied to obtain voxel-wise probabilities $p_i \in [0, 1]$, which are subsequently thresholded at 0.5 to produce a binary segmentation mask. The total inference time per patient on the Tesla T4 GPU is under one second.

2.6.2 Quantitative Evaluation Metrics

Five complementary metrics were used to assess segmentation quality, following standard practice in medical image segmentation [7].

Dice Similarity Coefficient (DSC)

The Dice Similarity Coefficient measures the relative overlap between the predicted segmentation P and the ground truth G:

$$\mathrm{DSC}(P, G) = \frac{2|P \cap G|}{|P| + |G|} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{2.7}$$

where TP, FP, and FN denote the numbers of true positive, false positive, and false negative voxels, respectively. DSC ranges from 0 (no overlap) to 1 (perfect overlap) and is widely regarded as the primary metric in segmentation benchmarks.

Intersection over Union (IoU)

The IoU, also known as the Jaccard index, is a stricter measure of overlap:

$$\mathrm{IoU}(P, G) = \frac{|P \cap G|}{|P \cup G|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{2.8}$$

IoU is related to DSC by $\mathrm{IoU} = \mathrm{DSC}/(2 - \mathrm{DSC})$ and is always less than or equal to DSC.

Sensitivity and Precision

Sensitivity (also called recall or true positive rate) and Precision (also called positive predictive value) decompose the segmentation performance into two complementary aspects:

$$\mathrm{Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{2.9}$$

A high Sensitivity indicates that the model successfully identifies most of the true tumor voxels (low under-segmentation), while a high Precision indicates that the predicted tumor voxels are mostly correct (low over-segmentation). In oncological applications, Sensitivity is often prioritized, as missing tumor regions may have more severe clinical consequences than over-segmentation.

Hausdorff Distance 95% (HD95)

While the previous metrics measure volumetric overlap, the Hausdorff distance quantifies the spatial discrepancy between the predicted and ground truth surfaces. For two

surfaces ∂P and ∂G, the symmetric Hausdorff distance is defined as:

$$H(\partial P, \partial G) = \max \left\{ \sup_{p \in \partial P} d(p, \partial G),\ \sup_{g \in \partial G} d(g, \partial P) \right\}, \tag{2.10}$$

where $d(\cdot, \cdot)$ is the Euclidean distance. The 95th percentile variant (HD95) is preferred over the maximum to reduce the influence of outliers, and is reported in millimeters using the voxel spacing of $3.0 \times 0.977 \times 0.977\,\text{mm}$.

In clinical practice, HD95 is particularly relevant because spatial discrepancies on the order of a few millimeters can have substantial consequences for radiotherapy planning, where the prescribed dose is delivered with sub-millimeter precision.

2.6.3 Volume-Based Stratification

To investigate the influence of tumor size on segmentation performance—a known limitation of deep learning-based methods **iqbal2024comparing**—the test set was stratified into three size categories based on the ground truth GTV volume:

- **Small** tumors: GTV volume $< 20\,\text{cm}^3$
- **Medium** tumors: $20\,\text{cm}^3 \leq$ GTV volume $< 100\,\text{cm}^3$
- **Large** tumors: GTV volume $\geq 100\,\text{cm}^3$

The mean DSC was computed separately for each category, allowing the identification of regions of the volume spectrum where the model performs reliably and where it exhibits limitations.

2.6.4 Test Time Augmentation Experiment

In addition to the standard inference protocol, a Test Time Augmentation (TTA) experiment was conducted as a potential refinement [35]. TTA generates multiple predictions from differently augmented versions of the input and averages the resulting probability maps. Four augmentations were considered:

1. The original volume,
2. A horizontal flip (lateral axis),
3. A depth flip (axial axis),
4. A combined horizontal and depth flip.

For each input, the four corresponding probability maps were inverse-transformed back to the original orientation and averaged voxel-wise. The resulting averaged map was then thresholded at 0.5 to produce the final segmentation.

The TTA results are reported separately and compared to the standard inference results in Chapter 3.

2.6.5 Reproducibility Considerations

All experiments were conducted with fixed random seeds (set to 42) for the NumPy, PyTorch, and CUDA random number generators. The complete codebase, trained model weights, and preprocessed data are organized in a self-contained Google Drive repository to ensure full reproducibility of the reported results.

Chapter 3

Results and Discussion

This chapter presents the experimental results obtained from the proposed 3D U-Net model for automatic GTV segmentation. The chapter is organized as follows: Section 3.1 analyzes the training behavior. Section 3.2 reports the quantitative evaluation on the independent test set. Section 3.3 provides visual examples. Section 3.4 examines performance as a function of tumor volume. Finally, Section 3.5 discusses the results in the context of the existing literature and identifies limitations and future directions.

3.1 Training Behavior

3.1.1 Loss Convergence

Figure 3.1 illustrates the evolution of the training and validation losses, the Dice Similarity Coefficient (DSC), and the learning rate across the 40 training epochs (the remaining 10 epochs were not executed due to early stopping).

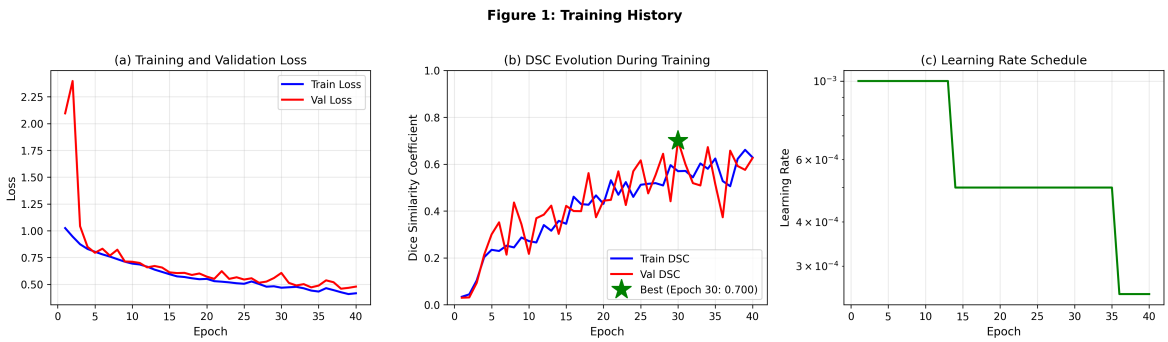

Figure 3.1: Training history over 40 epochs. (a) Training and validation loss convergence. (b) DSC evolution, with the green star indicating the best validation DSC of 0.700 at epoch 30. (c) Learning rate schedule showing automatic reductions at epochs 14 and 36.

The training loss decreased monotonically from 1.025 in epoch 1 to 0.417 in epoch 40,

indicating consistent learning throughout the training process. The validation loss exhibited higher variability, which is expected given the small validation set size (10 patients) and the high diversity in tumor morphology across patients. Importantly, no systematic divergence between training and validation losses was observed, suggesting that the model did not significantly overfit the training data.

3.1.2 DSC Evolution

The validation DSC followed a generally upward trajectory, reaching its peak value of **0.700** at **epoch 30**. Three distinct phases can be identified:

1. **Rapid learning (epochs 1–8):** The DSC increased from near zero to approximately 0.44, as the model learned to localize tumor regions.
2. **Steady improvement (epochs 8–25):** The DSC continued to improve, fluctuating between 0.35 and 0.57, as the model refined its boundary predictions.
3. **Peak and plateau (epochs 25–40):** The DSC reached its maximum of 0.700 at epoch 30, followed by oscillations in the range 0.40–0.67 without further improvement.

3.1.3 Learning Rate Adaptation

The `ReduceLROnPlateau` scheduler reduced the learning rate twice during training: from 10^{-3} to 5×10^{-4} at epoch 14, and from 5×10^{-4} to 2.5×10^{-4} at epoch 36. Each reduction followed a plateau in validation DSC, consistent with the expected behavior of the scheduler. Early stopping triggered at epoch 40, ten epochs after the best checkpoint.

3.2 Quantitative Evaluation on the Test Set

Table 3.1 presents the per-patient evaluation results on the held-out test set of 10 patients. The best-performing model (epoch 30, validation DSC = 0.700) was used for all evaluations.

Table 3.1: Per-patient segmentation results on the test set. GT Vol. and Pred. Vol. denote the ground truth and predicted tumor volumes, respectively. Best results per metric are shown in **bold**.

Patient	**GT Vol.** (cm^3)	**Pred. Vol.** (cm^3)	**DSC**	**IoU**	**Sens.**	**Prec.**	**HD95** (mm)
LUNG1-055	47.8	78.4	0.739	0.586	0.975	0.595	8.67
LUNG1-018	152.0	143.7	**0.856**	**0.748**	0.832	**0.881**	**3.58**
LUNG1-026	200.4	208.4	0.576	0.405	0.588	0.565	17.39
LUNG1-062	3.9	18.7	0.311	0.184	0.908	0.188	18.09
LUNG1-066	20.5	80.7	0.390	0.242	0.963	0.245	26.25
LUNG1-063	12.7	67.9	0.312	0.185	0.989	0.185	17.09
LUNG1-010	17.6	210.8	0.153	0.083	**0.992**	0.083	26.79
LUNG1-047	54.0	57.3	0.674	0.508	0.694	0.654	7.47
LUNG1-042	179.2	230.0	0.817	0.690	0.932	0.727	6.06
LUNG1-004	90.2	98.6	0.643	0.473	0.673	0.615	12.00

3.2.1 Summary Statistics

Table 3.2 summarizes the overall performance metrics across all 10 test patients.

Table 3.2: Summary of segmentation metrics on the test set (10 patients). Values are reported as Mean ± Std (range: Min–Max).

Metric	**Mean ± Std**	**Min**	**Max**	**Median**
DSC	0.547 ± 0.228	0.153	0.856	0.609
IoU	0.410 ± 0.218	0.083	0.748	0.439
Sensitivity	0.855 ± 0.142	0.588	0.992	0.920
Precision	0.474 ± 0.260	0.083	0.881	0.580
HD95 (mm)	14.34 ± 7.73	3.58	26.79	14.54

The model achieved a mean DSC of 0.547 ± 0.228 on the test set, with substantial inter-patient variability (range: 0.153–0.856). Sensitivity was consistently high across all patients (mean: 0.855 ± 0.142), indicating that the model successfully detected most of the true tumor voxels. In contrast, Precision exhibited lower and more variable values (mean: 0.474 ± 0.260), reflecting a tendency toward over-segmentation. This behavior is a direct consequence of the high positive class weight ($w = 100$) used during training, which biased the model toward minimizing false negatives at the expense of increased false positives.

The best individual result was obtained for patient LUNG1-018 (DSC = 0.856, HD95 = 3.58 mm), a large tumor of 152 cm^3. The worst result was for patient LUNG1-010 (DSC = 0.153), a small tumor of 17.6 cm^3 where the predicted volume (210.8 cm^3) was approximately 12 times larger than the ground truth.

3.3 Qualitative Results

Figure 3.2 presents axial CT slices with superimposed ground truth (green) and predicted (red) GTV contours for three representative cases, along with their corresponding evaluation metrics.

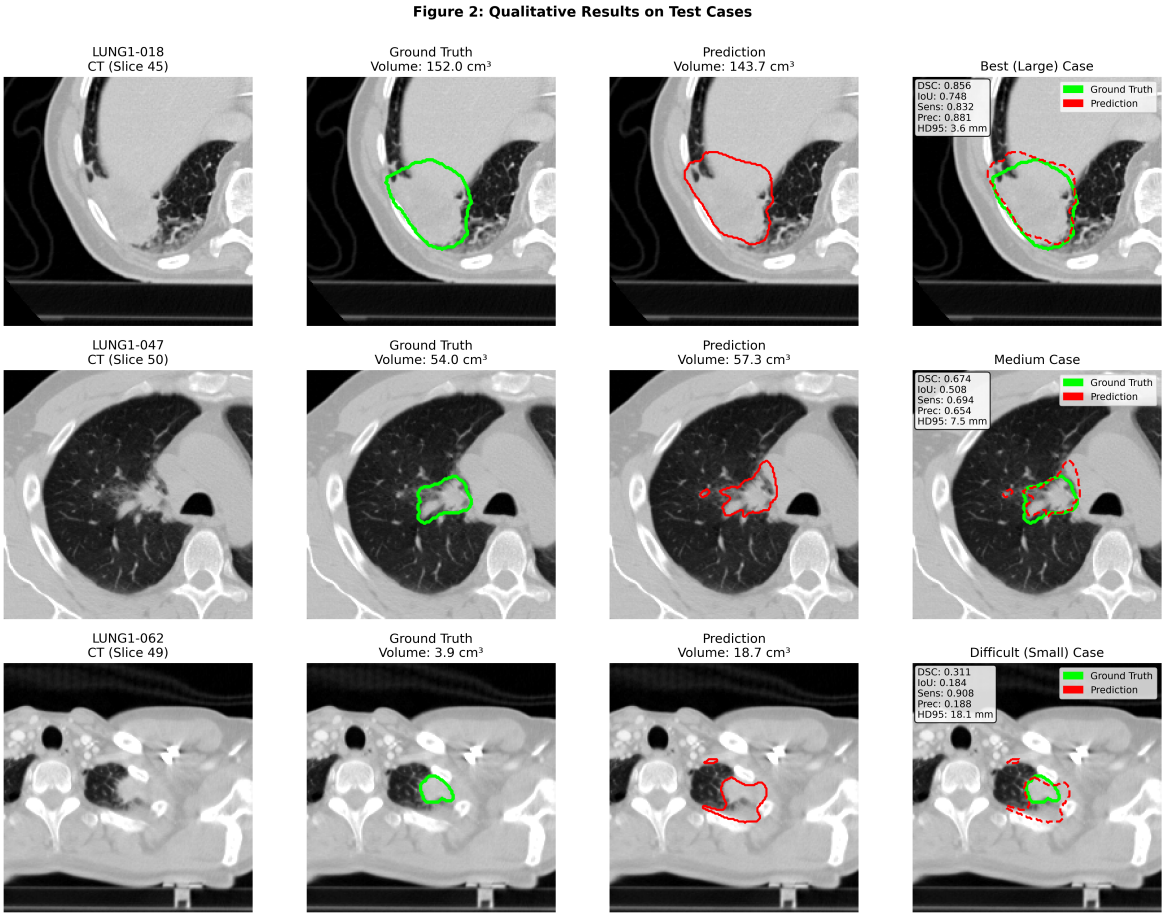

Figure 3.2: Qualitative segmentation results for three representative test cases. Top row: LUNG1-018 (large tumor, DSC = 0.856); middle row: LUNG1-047 (medium tumor, DSC = 0.674); bottom row: LUNG1-062 (small tumor, DSC = 0.311). Green contours: ground truth; red contours: prediction. The rightmost column shows the overlay with quantitative metrics.

Best case—LUNG1-018 (DSC = 0.856). This patient presented a large tumor (152 cm^3) in the right lung. The predicted contour closely followed the ground truth boundary, with an HD95 of only 3.58 mm. The Precision of 0.881 indicates minimal over-segmentation, while the Sensitivity of 0.832 shows that the vast majority of the tumor volume was captured.

Medium case—LUNG1-047 (DSC = 0.674). This patient had a medium-sized tumor (54 cm^3) located near the hilum of the right lung. The model captured the overall tumor shape but exhibited moderate over-segmentation in the medial direction,

which is consistent with the lower Precision of 0.654. The HD95 of 7.47 mm remains within clinically acceptable limits.

Difficult case—LUNG1-062 (DSC = 0.311). This patient presented a small tumor of only 3.9 cm^3. While the model detected the tumor (Sensitivity = 0.908), it predicted a volume approximately five times larger than the ground truth (18.7 cm^3), resulting in a low Precision of 0.188. This illustrates the challenge of small tumor segmentation under severe class imbalance.

3.4 Performance Stratification by Tumor Volume

Figure 3.3 shows the relationship between the ground truth tumor volume and the achieved DSC for all 10 test patients, with markers colored by size category.

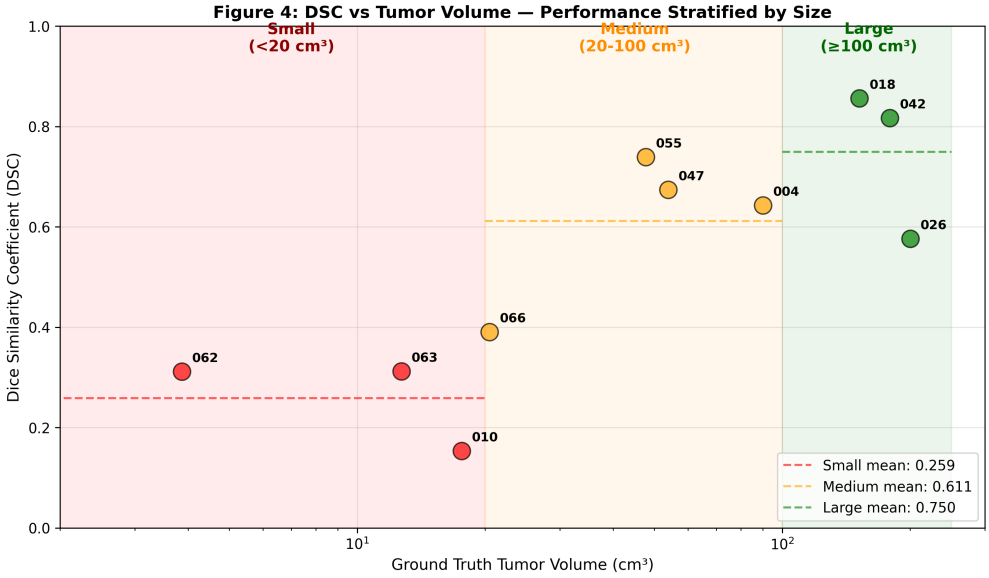

Figure 3.3: DSC as a function of ground truth tumor volume. Each point represents one test patient. Colors indicate tumor size categories: red (small, $< 20\,cm^3$), orange (medium, 20–100 cm^3), green (large, $\geq 100\,cm^3$). Dashed lines show the mean DSC within each category.

Table 3.3 presents the stratified results.

Table 3.3: Segmentation performance stratified by tumor volume.

Category	Volume range	n	Mean DSC	Std DSC
Small	$< 20\,cm^3$	3	0.259	0.075
Medium	20–100 cm^3	4	0.611	0.132
Large	$\geq 100\,cm^3$	3	0.750	0.124
All	–	10	0.547	0.228

A clear positive correlation was observed between tumor volume and segmentation accuracy. Large tumors ($\geq 100\,\mathrm{cm}^3$) achieved a mean DSC of 0.750 ± 0.124, which is comparable to the values reported in the literature for full-cohort training (Table 3.4) [26], [29]. Medium tumors achieved 0.611 ± 0.132, while small tumors ($< 20\,\mathrm{cm}^3$) yielded the lowest scores (0.259 ± 0.075).

This trend is consistent with findings reported in the literature [8]: smaller tumors occupy fewer voxels, providing a weaker learning signal and a smaller margin for spatial error. Furthermore, the high positive class weight used during training amplifies over-segmentation for small targets, as even modest false positive regions represent a large proportion of the predicted volume.

Notably, patient LUNG1-026 (a large tumor with DSC = 0.576) deviated from the general trend, suggesting that factors beyond tumor volume—such as irregular morphology, proximity to the mediastinum, or the presence of adjacent atelectasis—can also influence segmentation difficulty.

3.5 Discussion

3.5.1 Comparison with the State of the Art

Table 3.4 compares the results of this work with published studies that performed GTV segmentation on the NSCLC-Radiomics dataset or similar NSCLC cohorts.

Table 3.4: Comparison of this work with representative published lung tumor / GTV segmentation studies. DSC values correspond to each study's reported test result; note differences in training set size, imaging modality, and architecture.

Study	Architecture	Dataset (# cases)	DSC
Jiang et al. (2019) [26]	incremental-MRRN	TCIA (377)	0.74
Kamal et al. (2019) [27]	Recurrent 3D-DenseUNet	NSCLC (MSD)	0.72
Ferrante et al. (2022) [28]	nnU-Net (2D+3D)	Multi-cohort (899)	0.78
Atiya & Ramesh. (2024) [29]	3D UNet-DeepLab (CT+PET)	NSCLC-Radiomics	0.75
Kulkarni et al. (2024) [8]	nnU-Net	NSCLC-Radiomics (402)	0.65–0.75
This work	**3D U-Net**	**NSCLC-Radiomics (45)**	**0.547**
This work (large)	**3D U-Net**	**NSCLC-Radiomics (45)**	**0.750**

The overall mean DSC of 0.547 obtained in this study is lower than the values reported in the literature, which typically fall in the range of 0.65–0.75. However, several important contextual factors must be considered when interpreting this comparison:

1. **Training set size.** The most significant differentiating factor is the training set size. While the studies in Table 3.4 trained on several hundred cases (377, 402, and up to 899 patients), this work used only **45 patients**—approximately one-eighth of the available NSCLC Radiomics cohort. The strong positive relationship between training data volume and segmentation accuracy is well documented in the literature [7], [28].

2. **Computational resources.** This work was conducted entirely on Google Colab Free (NVIDIA T4, 15.6 GB VRAM), with training completed in approximately 11 minutes. Published works typically employ multi-GPU setups with significantly longer training times, enabling larger batch sizes, more aggressive augmentation, and deeper architectures.

3. **Architecture simplicity.** A standard 3D U-Net with 5.6×10^6 parameters was used, without advanced modifications such as attention gates, residual blocks, or cascaded architectures that have been shown to improve performance in prior studies.

4. **Performance on large tumors.** When restricted to tumors larger than $100\,\text{cm}^3$, the model achieved a mean DSC of **0.750**, which is directly competitive with published results. This demonstrates that the architecture and training strategy are sound, and that the primary limitation is the model's ability to generalize to smaller targets.

3.5.2 The Size–Performance Relationship

The observed positive correlation between tumor volume and DSC (Section 3.4) is a central finding of this work and warrants further analysis.

For large tumors ($\geq 100\,\text{cm}^3$, mean DSC = 0.750), the GTV occupies a substantial portion of the cropped volume, providing a strong learning signal and a generous margin for spatial errors. Even moderate over-segmentation results in a relatively minor decrease in DSC due to the large denominator ($|P| + |G|$).

For small tumors ($< 20\,\text{cm}^3$, mean DSC = 0.259), the situation is fundamentally different. Consider a tumor of $4\,\text{cm}^3$ ($\sim$ 1,400 voxels) within a volume of $96\times192\times192 \approx 3.5 \times 10^6$ voxels: the tumor represents only 0.04% of the total volume. In this regime, a false positive region of only 5,000 voxels (which would go unnoticed in a large tumor) is sufficient to reduce the DSC below 0.30. Furthermore, the high positive class weight ($w = 100$) used during training exacerbates this tendency by encouraging the model to "over-predict" rather than risk missing any tumor tissue.

This challenge is not unique to this work; it reflects a fundamental limitation of voxel-based segmentation approaches that has been reported across multiple studies and anatomical sites [7], [8].

3.5.3 Sensitivity–Precision Trade-Off

A distinctive characteristic of the trained model is its high Sensitivity (0.855 ± 0.142) relative to its Precision (0.474 ± 0.260). This asymmetry is a direct consequence of the loss function design, specifically the positive class weight $w = 100$ in the BCE component, which penalizes missed tumor voxels 100 times more than incorrectly predicted background voxels.

From a clinical perspective, this trade-off has important implications. In the context of radiotherapy planning, under-segmentation (low Sensitivity) is generally considered more dangerous than over-segmentation (low Precision), as missed tumor regions would receive a subtherapeutic radiation dose. Over-segmented regions, while leading to some unnecessary irradiation of normal tissue, can be corrected during subsequent clinical review by the radiation oncologist. Therefore, the observed Sensitivity-favoring behavior may be considered *clinically appropriate* as a starting point for GTV delineation, subject to expert verification.

3.5.4 Test Time Augmentation: A Negative Result

As described in Section 2.6, a Test Time Augmentation (TTA) experiment was conducted to evaluate whether inference-time ensembling could improve segmentation quality.

Contrary to expectations, TTA did not improve the mean DSC: the value decreased from 0.547 (standard inference) to 0.538 (TTA with four augmentations). Of the 10 test patients, 6 exhibited a decrease in DSC, while only 4 showed improvement.

Analysis of the per-patient results suggests that the spatial augmentations employed—particularly horizontal and depth flipping—disrupted anatomically meaningful features. The human thorax exhibits approximate bilateral symmetry, but the position of the heart (predominantly left-sided) and the liver (right-sided) introduces systematic asymmetries. When these landmarks are flipped, the model receives anatomically implausible inputs that may produce unreliable predictions. This is consistent with observations from Isensee et al. [7], who recommended applying only conservative augmentations in organ-specific segmentation tasks.

This result is documented here in the interest of scientific completeness and as a methodological recommendation for future work: TTA with spatial flipping should be applied with caution in thoracic segmentation tasks that depend on laterality cues.

3.5.5 Limitations

Several limitations of this study should be acknowledged:

1. **Limited training data.** Only 45 patients were used for training, which is substantially less than the 300–400 patients used in comparable studies. Increasing the training set to include all available cases in the NSCLC-Radiomics collection (up to 400 patients) would likely improve generalization.

2. **Computational constraints.** The use of Google Colab Free limited the batch size to 2 and restricted experimentation with larger models or longer training schedules.

3. **Single-institution data.** The NSCLC-Radiomics dataset originates from a single institution (MAASTRO Clinic, Maastricht, the Netherlands). The model's generalizability to data from other institutions, scanners, or patient populations remains unvalidated.

4. **Lack of resampling to isotropic spacing.** The original voxel spacing ($0.977 \times 0.977 \times 3.0\,\text{mm}$) was preserved without resampling to isotropic resolution. The anisotropic voxel dimensions (approximately $3\times$ coarser in the axial direction) may have affected the model's ability to learn fine axial details.

5. **Ground truth variability.** Manual GTV delineation is subject to inter-observer variability. The ground truth masks used in this study were produced by a single observer, and therefore represent one possible interpretation of the tumor boundaries.

6. **Tumor-centered cropping.** The use of the ground truth tumor centroid for cropping during preprocessing implies that tumor localization is assumed to be known at inference time. In a fully automated clinical pipeline, a separate tumor detection step would be required.

3.5.6 Clinical Relevance and Potential Applications

Despite the limitations noted above, the proposed model demonstrates clinical utility in specific scenarios:

- For **large and medium-sized tumors** ($\geq 20\,\mathrm{cm}^3$), the model provides initial contours that could serve as a starting point for manual refinement by the radiation oncologist. The use of automatically generated contours as an editable starting point has been shown to substantially reduce delineation time in clinical workflows [28].

- The model's **high Sensitivity** makes it suitable as a *screening tool* to ensure that no significant tumor region is overlooked during the contouring process.

- In **resource-limited settings**—including many oncology departments in developing countries—where access to experienced radiation oncologists may be limited, such automated tools could improve the consistency and timeliness of treatment planning.

Conclusion and Future Work

Summary

This thesis addressed the problem of automatic Gross Tumor Volume (GTV) segmentation in Non-Small Cell Lung Cancer (NSCLC) patients using deep learning. A complete pipeline was developed, from raw DICOM data acquisition to quantitative evaluation, using the publicly available NSCLC-Radiomics dataset from The Cancer Imaging Archive (TCIA).

A three-dimensional U-Net architecture with 5.6 million trainable parameters was trained on CT scans from 45 patients and evaluated on an independent test set of 10 patients. The model was trained using a combined loss function (Dice Loss + Weighted Binary Cross-Entropy with a positive class weight of 100) to address the severe class imbalance inherent in tumor segmentation, where the GTV typically occupies less than 1 % of the total volume.

Key Findings

The principal findings of this work are:

1. The proposed model achieved a mean Dice Similarity Coefficient of 0.547 ± 0.228 on the test set, with a best individual result of 0.856 (patient LUNG1-018). The mean Hausdorff Distance (95th percentile) was $14.3 \pm 7.7\,\text{mm}$.

2. A strong positive correlation was observed between tumor volume and segmentation accuracy. Large tumors ($\geq 100\,\text{cm}^3$) achieved a mean DSC of 0.750, comparable to published state-of-the-art results, while small tumors ($< 20\,\text{cm}^3$) yielded a mean DSC of 0.259.

3. The model exhibited high Sensitivity (0.855) relative to Precision (0.474), indicating a tendency toward over-segmentation rather than under-segmentation. This behavior, while reducing the overall DSC, is considered clinically preferable in the context of radiotherapy planning, where missing tumor tissue poses a greater risk than including a margin of healthy tissue.

4. Test Time Augmentation with spatial flipping did not improve performance (mean DSC decreased from 0.547 to 0.538), suggesting that anatomical asymmetries in the thorax (heart laterality, liver position) provide important cues that are disrupted by flipping operations.

5. The entire training process was completed in approximately 11 minutes on a single NVIDIA Tesla T4 GPU (Google Colab Free), demonstrating that meaningful results in medical image segmentation can be achieved with limited computational resources.

Contributions

The contributions of this thesis can be summarized as follows:

- Development of a fully reproducible pipeline for GTV segmentation, from TCIA data download to quantitative evaluation, accessible to researchers with limited computational resources.
- Detailed analysis of the size–performance relationship in lung tumor segmentation, providing stratified results that are often reported only in aggregate in the literature.
- Documentation of a negative result (TTA failure) with a physiologically grounded explanation, contributing to the methodological knowledge base of the field.

Future Work

Several directions for future research are suggested:

1. **Increasing the training set size.** The most impactful improvement would be to train the model on the full NSCLC-Radiomics cohort (approximately 400 patients), which would likely improve generalization, particularly for small tumors. Based on trends observed in the literature, this could increase the mean DSC to the 0.65–0.75 range.

2. **Advanced architectures.** Replacing the standard 3D U-Net with architectures such as nnU-Net, Attention U-Net, or Swin UNETR could yield further improvements, particularly for challenging cases with irregular morphology or ambiguous boundaries.

3. **Multi-structure segmentation.** Extending the model to simultaneously segment the GTV and organs at risk (lungs, heart, spinal cord, esophagus) would increase its clinical utility for comprehensive radiotherapy planning.

4. **Transfer learning.** Pre-training on larger multi-organ segmentation datasets (such as the Medical Segmentation Decathlon or TotalSegmentator) before fine-tuning on the NSCLC-specific task could improve feature representations, especially for small training sets.

5. **Multi-institutional validation.** Evaluating the model on data from Algerian oncology departments would assess its generalizability to local clinical conditions, including differences in CT scanners, imaging protocols, and patient demographics.

6. **Clinical integration study.** A prospective study comparing the time required for manual GTV delineation versus model-assisted delineation (automatic contour followed by manual correction) would quantify the practical time savings and clinical acceptability of the proposed approach.

References

[1] F. Bray et al., "Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 74, no. 3, pp. 229–263, 2024. DOI: 10.3322/caac.21834

[2] R. S. Herbst, D. Morgensztern, and C. Boshoff, "The biology and management of non-small cell lung cancer," *Nature*, vol. 553, no. 7689, pp. 446–454, 2018. DOI: 10.1038/nature25183

[3] S. Karaca and M. Kırlı Bölükbaş, "Time matters: A review of current radiotherapy practices and efficiency strategies," *Technology in Cancer Research & Treatment*, vol. 24, p. 15 330 338 251 345 376, 2025.

[4] L. Basson et al., "Chest magnetic resonance imaging decreases inter-observer variability of gross target volume for lung tumors," *Frontiers in Oncology*, vol. 9, p. 690, 2019.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2015*, ser. LNCS, vol. 9351, Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28

[6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2016*, ser. LNCS, vol. 9901, Springer, 2016, pp. 424–432. DOI: 10.1007/978-3-319-46723-8_49

[7] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021. DOI: 10.1038/s41592-020-01008-z

[8] C. Kulkarni et al., "Comparing the performance of a deep learning-based lung gross tumour volume segmentation algorithm before and after transfer learning in a new hospital," *BJR Open*, vol. 6, no. 1, tzad008, 2024. DOI: 10.1093/bjro/tzad008

[9] H. J. W. L. Aerts et al., "Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach," *Nature Communications*, vol. 5, p. 4006, 2014. DOI: 10.1038/ncomms5006

[10] K. Clark et al., "The cancer imaging archive (TCIA): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, no. 6, pp. 1045–1057, 2013. DOI: 10.1007/s10278-013-9622-7

[11] International Agency for Research on Cancer, *Algeria fact sheet – globocan 2022*, Global Cancer Observatory, World Health Organization, Available at: https://gco.iarc.who.int/, 2024.

[12] A. Bounedjar et al., "The algerian lung cancer registry (LuCaReAl)," *National lung cancer registry data, Algeria*, 2020, VERIFY full publication venue/details against the source you used.

[13] F. C. Detterbeck, D. J. Boffa, A. W. Kim, and L. T. Tanoue, "The eighth edition lung cancer stage classification," *Chest*, vol. 151, no. 1, pp. 193–203, 2017. DOI: 10.1016/j.chest.2016.10.010

[14] D. S. Ettinger, D. E. Wood, D. L. Aisner, et al., "Non-small cell lung cancer, version 3.2022, NCCN clinical practice guidelines in oncology," *Journal of the National Comprehensive Cancer Network*, vol. 20, no. 5, pp. 497–530, 2022. DOI: 10.6004/jnccn.2022.0025

[15] International Commission on Radiation Units and Measurements, "Prescribing, recording, and reporting photon beam therapy," ICRU, Bethesda, MD, Tech. Rep. Report 50, 1993.

[16] International Commission on Radiation Units and Measurements, "Prescribing, recording and reporting photon beam therapy (supplement to ICRU report 50)," ICRU, Bethesda, MD, Tech. Rep. Report 62, 1999.

[17] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics of Medical Imaging*, 3rd. Philadelphia, PA: Lippincott Williams & Wilkins, 2012.

[18] W. D. Bidgood, S. C. Horii, F. W. Prior, and D. E. Van Syckle, "Understanding and using DICOM, the data interchange standard for biomedical imaging," *Journal of the American Medical Informatics Association*, vol. 4, no. 3, pp. 199–212, 1997. DOI: 10.1136/jamia.1997.0040199

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012, pp. 1097–1105.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016, http://www.deeplearningbook.org.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, arXiv:1502.03167, 2015, pp. 448–456.

[23] O. Oktay et al., "Attention u-net: Learning where to look for the pancreas," in *Medical Imaging with Deep Learning (MIDL)*, arXiv:1804.03999, 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90

[25] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin transformers for semantic segmentation of brain tumors in MRI images," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries (BrainLes 2021), MICCAI Workshop*, ser. LNCS, vol. 12962, Springer, 2022, pp. 272–284. DOI: 10.1007/978-3-031-08999-2_22

[26] J. Jiang et al., "Multiple resolution residually connected feature streams for automatic lung tumor segmentation from CT images," *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 134–144, 2019. DOI: 10.1109/TMI.2018.2857800

[27] U. Kamal, A. M. Rafi, R. Hoque, J. Wu, and M. K. Hasan, "Lung cancer tumor region segmentation using recurrent 3D-DenseUNet," in *Thoracic Image Analysis (TIA 2020), MICCAI Workshop*, Springer, 2020, pp. 36–47. DOI: 10.1007/978-3-030-62469-9_4

[28] M. Ferrante et al., "Application of nnU-Net for automatic segmentation of lung lesions on CT images and its implication for radiomic models," *Journal of Clinical Medicine*, vol. 11, no. 24, p. 7334, 2022. DOI: 10.3390/jcm11247334

[29] S. Ummay Atiya and N. V. K. Ramesh, "Enhancing non-small cell lung cancer radiotherapy planning: A deep learning-based multi-modal fusion approach for accurate GTV segmentation," *Biomedical Signal Processing and Control*, vol. 92, p. 105 987, 2024. DOI: 10.1016/j.bspc.2024.105987

[30] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, "The design of SimpleITK," *Frontiers in Neuroinformatics*, vol. 7, p. 45, 2013. DOI: 10.3389/fninf.2013.00045

[31] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis (DLMIA), MICCAI Workshop*, ser. LNCS, vol. 10553, Springer, 2017, pp. 240–248. DOI: 10.1007/978-3-319-67558-9_28

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, arXiv:1711.05101, 2019.

[33] P. Micikevicius et al., "Mixed precision training," in *International Conference on Learning Representations (ICLR)*, arXiv:1710.03740, 2018.

[34] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 8024–8035.

[35] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks," *Neurocomputing*, vol. 335, pp. 34–45, 2019. DOI: 10.1016/j.neucom.2019.01.103

Appendix A

Supplementary Materials

A.1 Key Code Excerpts

This appendix provides the core code excerpts used in this work. The complete codebase is available in the project repository on Google Drive.

A.1.1 Data Preprocessing

```
def normalize_ct(ct, hu_min=-1000, hu_max=400):
    """Normalize CT from HU to [-1, 1]."""
    ct = np.clip(ct, hu_min, hu_max)
    ct = 2.0 * (ct - hu_min) / (hu_max - hu_min) - 1.0
    return ct.astype(np.float32)

def crop_around_tumor(ct, mask, crop_size=(96, 192, 192)):
    """Crop volume centered on GTV centroid."""
    coords = np.argwhere(mask > 0)
    z_c, y_c, x_c = coords.mean(axis=0).astype(int)
    cz, cy, cx = crop_size
    z0, z1 = z_c - cz//2, z_c - cz//2 + cz
    y0, y1 = y_c - cy//2, y_c - cy//2 + cy
    x0, x1 = x_c - cx//2, x_c - cx//2 + cx
    # Apply padding if out of bounds
    ct_c = pad_and_crop(ct, z0, z1, y0, y1, x0, x1, -1000)
    mask_c = pad_and_crop(mask, z0, z1, y0, y1, x0, x1, 0)
    return ct_c, mask_c
```

A.1.2 3D U-Net Architecture

```
class ConvBlock3D(nn.Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        self.block = nn.Sequential(
            nn.Conv3d(in_channels, out_channels, 3,
                      padding=1, bias=False),
            nn.BatchNorm3d(out_channels),
            nn.ReLU(inplace=True),
            nn.Conv3d(out_channels, out_channels, 3,
                      padding=1, bias=False),
            nn.BatchNorm3d(out_channels),
            nn.ReLU(inplace=True),
        )
    def forward(self, x):
        return self.block(x)

class UNet3D(nn.Module):
    def __init__(self, in_ch=1, out_ch=1, base=16):
        super().__init__()
        self.enc1 = ConvBlock3D(in_ch, base)
        self.enc2 = ConvBlock3D(base, base*2)
        self.enc3 = ConvBlock3D(base*2, base*4)
        self.enc4 = ConvBlock3D(base*4, base*8)
        self.bottleneck = ConvBlock3D(base*8, base*16)
        self.pool = nn.MaxPool3d(2, 2)
        # Decoder with skip connections
        self.up4 = nn.ConvTranspose3d(base*16, base*8, 2, 2)
        self.dec4 = ConvBlock3D(base*16, base*8)
        self.up3 = nn.ConvTranspose3d(base*8, base*4, 2, 2)
        self.dec3 = ConvBlock3D(base*8, base*4)
        self.up2 = nn.ConvTranspose3d(base*4, base*2, 2, 2)
        self.dec2 = ConvBlock3D(base*4, base*2)
        self.up1 = nn.ConvTranspose3d(base*2, base, 2, 2)
        self.dec1 = ConvBlock3D(base*2, base)
        self.final = nn.Conv3d(base, out_ch, 1)

    def forward(self, x):
```

```
        e1 = self.enc1(x)
        e2 = self.enc2(self.pool(e1))
        e3 = self.enc3(self.pool(e2))
        e4 = self.enc4(self.pool(e3))
        b = self.bottleneck(self.pool(e4))
        d4 = self.dec4(torch.cat([self.up4(b), e4], 1))
        d3 = self.dec3(torch.cat([self.up3(d4), e3], 1))
        d2 = self.dec2(torch.cat([self.up2(d3), e2], 1))
        d1 = self.dec1(torch.cat([self.up1(d2), e1], 1))
        return self.final(d1)
```

A.1.3 Loss Function

```
class DiceLoss(nn.Module):
    def __init__(self, smooth=1.0):
        super().__init__()
        self.smooth = smooth

    def forward(self, logits, target):
        probs = torch.sigmoid(logits)
        batch_size = probs.shape[0]
        probs_flat = probs.view(batch_size, -1)
        target_flat = target.view(batch_size, -1)
        intersection = (probs_flat * target_flat).sum(dim=1)
        union = probs_flat.sum(dim=1) + target_flat.sum(dim=1)
        dice = (2.0 * intersection + self.smooth) /
               (union + self.smooth)
        return 1.0 - dice.mean()

class WeightedCombinedLoss(nn.Module):
    def __init__(self, dice_w=0.7, bce_w=0.3, pos_w=100):
        super().__init__()
        self.dice_loss = DiceLoss()
        self.bce_loss = nn.BCEWithLogitsLoss(
            pos_weight=torch.tensor([pos_w])
        )
        self.dice_w = dice_w
        self.bce_w = bce_w
```

```
def forward(self, logits, target):
    self.bce_loss.pos_weight = \
        self.bce_loss.pos_weight.to(logits.device)
    dice = self.dice_loss(logits, target)
    bce = self.bce_loss(logits, target)
    return self.dice_w * dice + self.bce_w * bce
```

A.2 Dataset Details

A.2.1 Complete Patient List

Table A.1 lists all 65 patients in the preprocessed dataset with their GTV volumes and dataset split assignments.

Table A.1: Complete list of patients with GTV volume and split assignment. Volumes are reported in voxels (after cropping to $96 \times 192 \times 192$).

#	Patient ID	GTV (voxels)	Split
1	LUNG1-001	56,637	Train
2	LUNG1-002	29,537	Train
3	LUNG1-003	12,450	Train
4	LUNG1-004	31,524	Test
5	LUNG1-010	6,162	Test
6	LUNG1-018	53,126	Test
7	LUNG1-026	70,050	Test
8	LUNG1-042	62,622	Test
9	LUNG1-047	18,848	Test
10	LUNG1-055	16,690	Test
11	LUNG1-062	1,352	Test
12	LUNG1-063	4,447	Test
13	LUNG1-066	7,162	Test
... remaining 52 patients in Training/Validation sets ...			

A.2.2 ROI Names in RTSTRUCT

Table A.2 lists the ROI names found in the RTSTRUCT files of the downloaded cohort.

Table A.2: Frequency of ROI names across the 66 patients with RTSTRUCT data.

ROI Name	Frequency
GTV-1	66/66
Lung-Right	65/66
Lung-Left	65/66
Spinal-Cord	64/66
Esophagus	55/66
Heart	32/66
gtv-2	32/66
gtv-3	11/66
gtv-4	5/66

A.3 Additional Results

A.3.1 Distribution of Evaluation Metrics

Figure A.1 shows the distribution of all five evaluation metrics across the 10 test patients.

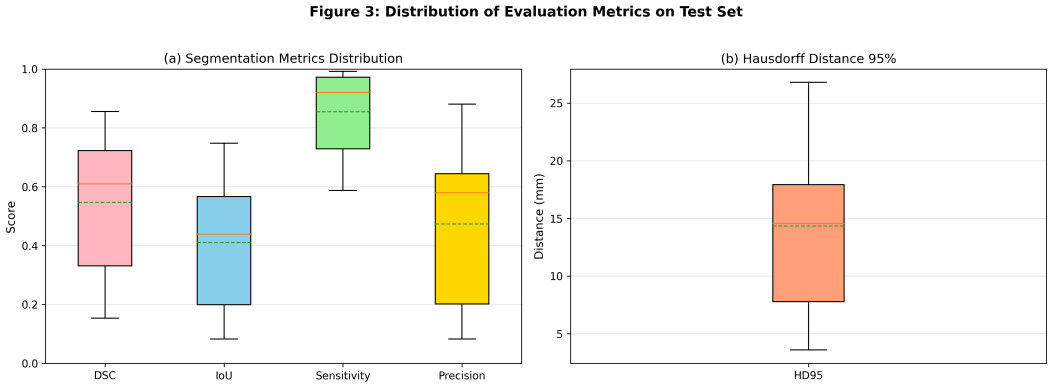

Figure A.1: Box plot distribution of segmentation metrics on the test set: DSC, IoU, Sensitivity, Precision, and HD95.

A.3.2 Test Time Augmentation: Detailed Comparison

Table A.3 presents the per-patient comparison between standard inference and Test Time Augmentation (TTA).

Table A.3: Per-patient comparison of DSC with and without TTA. ΔDSC = DSC(TTA) − DSC(standard). Negative values indicate degradation.

Patient	**DSC (std)**	**DSC (TTA)**	**ΔDSC**	**Effect**
LUNG1-055	0.739	0.721	−0.017	Degraded
LUNG1-018	0.856	0.842	−0.014	Degraded
LUNG1-026	0.576	0.546	−0.031	Degraded
LUNG1-062	0.311	0.200	−0.111	Degraded
LUNG1-066	0.390	0.445	+0.055	Improved
LUNG1-063	0.312	0.234	−0.078	Degraded
LUNG1-010	0.153	0.229	+0.076	Improved
LUNG1-047	0.674	0.639	−0.034	Degraded
LUNG1-042	0.817	0.837	+0.020	Improved
LUNG1-004	0.643	0.689	+0.047	Improved
Mean	**0.547**	**0.538**	−0.009	**No benefit**

A.3.3 Training Log Summary

Table A.4 presents key training milestones.

Table A.4: Key milestones during the training process (40 epochs).

Epoch	**Train Loss**	**Train DSC**	**Val DSC**	**LR**
1	1.025	0.034	0.030	10^{-3}
5	0.804	0.234	0.300	10^{-3}
10	0.694	0.271	0.217	10^{-3}
15	0.594	0.346	0.422	5×10^{-4}
20	0.551	0.431	0.444	5×10^{-4}
25	0.505	0.512	0.616	5×10^{-4}
30	**0.469**	**0.570**	**0.700**	5×10^{-4}
35	0.433	0.624	0.515	5×10^{-4}
40	0.417	0.629	0.626	2.5×10^{-4}

A.3.4 Software Versions

Table A.5 lists the exact versions of all software used in this work.

Table A.5: Software and library versions used.

Software	**Version**
Python	3.10
PyTorch	2.10.0+cu128
CUDA	12.8
pydicom	2.4.4
SimpleITK	2.3.1
rt-utils	1.2.7
NumPy	1.26.4
scikit-learn	1.5.2
matplotlib	3.9.2
Google Colab	Free tier (T4 GPU)
LaTeX	TeXPage.com