TU WIEN Informatics

Lung and Lung Cancer Segmentation using Deep Learning

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Valentin Milicevic, BSc
Matrikelnummer 12122084

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ. Prof. Dr. Allan Hanbury
Mitwirkung: Dipl. Ing. Philipp Seeboeck, PhD
Univ. Prof. Dipl. Ing. Georg Langs

Wien, 28. März 2024

Valentin Milicevic

Allan Hanbury

Informatics

Lung and Lung Cancer Segmentation using Deep Learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Valentin Milicevic, BSc
Registration Number 12122084

to the Faculty of Informatics

at the TU Wien

Advisor: Univ. Prof. Dr. Allan Hanbury
Assistance: Dipl. Ing. Philipp Seeboeck, PhD
Univ. Prof. Dipl. Ing. Georg Langs

Vienna, 28th March, 2024

Valentin Milicevic

Allan Hanbury

Erklärung zur Verfassung der Arbeit

Valentin Milicevic, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 28. März 2024

Valentin Milicevic

Acknowledgements

First and foremost, my deepest gratitude goes to Philipp Seeboeck from the Computational Imaging Research (CIR) Lab for his guidance and supervision through my thesis. His willingness to discuss and patiently answer all of my questions has been a key part of this work. I have learned and benefited greatly from your expertise.

My thanks to Georg Langs from the CIR Lab for giving me this fantastic opportunity to join your research team. His motivation and encouragement during our interdisciplinary project led me to pursue a thesis project with the CIR Lab. Also, his interesting research topic was truly inspiring. Many thanks to Allan Hanbury for his consistent support throughout my thesis and my entire studies at TU Wien. I appreciate your advice and fast responses in all situations.

I am thankful to my friends from Austria, Croatia, and around the world for their support during my stay in Vienna. Your friendship and motivation were always highly valuable to me.

Finally, my sincere thanks to my beloved family - Leonarda, Ante Dominik, Ozana, Tereza, and Drazen for supporting my decision to move to Vienna for studies in every possible way. Their sacrifices and unconditional love have always been the driving focus behind my achievements.

This work was conducted in collaboration with the Computational Imaging Research Lab (CIR) from the Medical University of Vienna, which is an interdisciplinary team of researchers with medicine, computer science, and mathematics background.

Kurzfassung

Lungenkrebs ist weltweit die häufigste und tödlichste Krebserkrankung sowohl bei Männern als auch bei Frauen und stellt die Diagnose und Behandlung vor große Herausforderungen. Die Ätiologie des Lungenkrebses, der überwiegend mit dem Rauchen in Verbindung gebracht wird, ist ein komplexer Prozess, der durch verschiedene Umweltfaktoren wie die Belastung durch Luftverschmutzung beeinflusst werden kann. Er wird grob in kleinzellige Lungenkarzinome (SCLC) und nicht-kleinzellige Lungenkarzinome (NSCLC) eingeteilt, die sich unterschiedlich ausbreiten und wachsen. Das Hauptproblem bei Lungenkrebs ist die späte Diagnose, da die Symptome in der Regel verzögert auftreten. Dieses fortgeschrittene Krankheitsstadium ist mit den derzeit verfügbaren Therapien nahezu unheilbar. Diese Therapien beruhen in erster Linie auf der manuellen Erkennung und Segmentierung von CT-Bildern durch Radiologen. Dieser manuelle Prozess hat jedoch mehrere Nachteile, darunter der hohe Zeitaufwand und die Schwierigkeit, die Tumorgröße präzise und zuverlässig zu quantifizieren.

Diese Arbeit untersucht das Potenzial von Deep Learning, insbesondere von Convolutional Neural Networks (CNNs) und Vision Transformers (ViTs), bei der Segmentierung von Lungenkrebs. In diesem Zusammenhang befassen wir uns auch mit der damit verbundenen Aufgabe der Lungensegmentierung. Wir schlagen die Verwendung von CNN-basierten Modellen und dem ViT-basierten Segment Anything Model (SAM) vor, das kürzlich von Meta AI mit verschiedenen Konfigurationen (z.B. Netzwerk-Encoder) veröffentlicht wurde. Das Hauptziel ist es, diese Modelle durch verschiedene Konfigurationen zu optimieren und ihre Leistung bei verschiedenen Segmentierungsaufgaben zu vergleichen.

Die wichtigsten Ergebnisse zeigen signifikante Unterschiede zwischen CNN-basierten und ViT-basierten Modellen in verschiedenen Modellkonfigurationen und Datensätzen. Da SAM menschliche Interaktion (sogenannte "Prompts") verwendet, haben wir die Auswirkungen von SAM mit und ohne simulierte menschliche Interaktion untersucht. Darüber hinaus haben wir die Leistung der CNN-basierten und ViT-basierten Modelle für drei verschiedene Tumorgrößen untersucht, darunter eine kleine, mittlere und große Größe. Im Allgemeinen zeigen unsere Ergebnisse vielversprechende Leistungsfähigkeiten von CNN-basierten und ViT-basierten Modellen sowohl für Lungen- als auch für Lungenkrebs-Segmentierungsaufgaben. CNN-basierte Modelle erreichen den höchsten Dice-Score von 0,975 mit U-Net mit efficientnet-b7 bei der Lungensegmentierung, verglichen mit nur 0,440 mit U-Net mit resnet101 bei der Segmentierung von Lungenkrebs. Bei der Lungenkrebs-

Segmentierungsaufgabe verbessert sich der Dice-Score jedoch drastisch auf einen Wert von 0,749 bei Verwendung von SAM mit simulierter menschlicher Interaktion. Diese Ergebnisse könnten bei der künftigen Entwicklung von Software für die Behandlung von Lungenkrebs im Frühstadium helfen, die auf präzisen Segmentierungsmodellen basiert, da Radiologen diese Modelle zur präzisen und effektiven Quantifizierung des Tumors verwenden könnten.

Abstract

Lung cancer, as the most prevalent and deadliest cancer globally among both men and women, presents significant diagnostic and treatment challenges. The etiology of lung cancer, predominately linked to smoking, is a complex process that may be influenced by various environmental factors such as air pollution exposure. It is broadly classified into small-cell lung carcinomas (SCLC) and non-small-cell lung carcinomas (NSCLC) that spread and grow differently. Lung cancer's major issue is late diagnosis since the symptoms are usually delayed. This advanced stage of the disease is almost incurable with currently available therapies. These therapies primarily rely on manual CT image detection and segmentation by radiologists. However, this process suffers from several limitations including its time-consuming nature and the difficulty to precisely and reliably quantify tumor size.

This thesis aims to explore the potential of deep learning, particularly Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) in lung cancer segmentation. In this context, we also tackle the associated task of lung segmentation. We propose the use of CNN-based models and ViT-based Segment Anything Model that was recently published by Meta AI with various configurations (e.g. network encoders). The primary objective is to optimize these models through various configurations and benchmark their performance across different segmentation tasks.

Key findings show significant differences between CNN-based and ViT-based models across different model configurations and datasets. Since SAM uses prompts, we explored the impact of SAM with and without simulated human interaction. In addition, we assessed the CNN-based and ViT-based model's performance for three different tumor scales including small, medium, and large scale. Generally, our results demonstrate promising capabilities of CNN-based and ViT-based models for both lung and lung cancer segmentation tasks. CNN-based models achieve the highest dice score of 0.975 using U-Net with efficientnet-b7 on the lung segmentation task compared to only 0.440 using U-Net with resnet101 on the lung cancer segmentation task. However, the dice score drastically improves to a dice score of 0.749 using SAM with simulated human interaction for the lung cancer segmentation task. These findings may help in the future development of early-stage lung cancer treatment software that is based on precise segmentation models since radiologists could use these models to quantify the tumor in a precise and effective manner.

Contents

CHAPTER 1

Introduction

Lung cancer is recognized as the most prevalent cancer and the leading cause of cancer-related mortality among both men and women, contributing to 18.4% of global cancer-related deaths in 2018 [BFS+18].

The major etiological factor in approximately 90% of lung cancer is attributed to smoking and tobacco product usage. However, other factors such as exposure to radon gas, asbestos, air pollution, and chronic infections may contribute to lung carcinogenesis. Various inherited and acquired mechanisms of susceptibility to lung cancer have been proposed. Lung cancer can be classified into two broad histological classes, which grow and spread differently: small-cell lung carcinomas (SCLC) and non-small-cell lung carcinomas (NSCLC). Treatment options for lung cancer encompass surgical intervention, radiation therapy, and targeted therapy. The choice of therapeutic modalities depends on multiple factors, including the cancer type and stage [LAHYB15].

The majority of patients with lung cancer are diagnosed with advanced disease due to the delayed manifestation of symptoms during the disease. Unfortunately for those patients, after the metastases occur, the disease is not curable with the currently available therapies [PKO+17].
Nowadays, the diagnosis of lung cancer relies on the manual detection and segmentation of CT images by radiation oncologists. This process suffers from several limitations, including inter and intra-observer variability, the likelihood of missing small cancer regions, its time-consuming nature, and the inability to provide precise quantification of tumors [PIVT+22a].
However, the implementation of deep learning based lung cancer segmentation models has the potential to help overcome these challenges. The integration of automated detection and segmentation techniques would have an immediate impact on the clinical workflow within radiotherapy, leading to more efficient and consistent lung cancer diagnosis [PKO+17][PIvT+22b][SASL23].

In the field of medical imaging, many state-of-the-art machine learning models have shown promising results in lung cancer segmentation, especially using Convolutional Neural Networks (CNNs). However, these current approaches still have limitations when it comes to generalizability, small lesion detection, speed, and efficiency.

To tackle this problem, we propose to implement both CNN-based models and the recently published Vision Transformer (ViT) based model Segment Anything (SAM) for lung cancer segmentation [KMR+23] [GWK+15]. The primary objective of this thesis is to benchmark these models using various configurations (e.g. backbones or model architectures). We also tackle the task of lung segmentation since it is often one of the crucial steps in lung cancer segmentation workflows that are implemented in clinical settings [PIVT+22a].

1.1 Research Questions

This thesis focuses on the following research questions:

1. *How does the CNN-based and ViT-based model performance vary across different datasets?*
   This research question aims to compare the CNN-based and ViT-based models' performance on two different datasets, or segmentation tasks, including lung segmentation and lung cancer segmentation. In this work, we only utilize SAM as our ViT-based model.

2. *What is the impact of simulated human interaction with SAM in terms of performance?*
   This research question assesses the impact of simulated human interaction for fine-tuned SAM compared to SAM without simulated human interaction.

3. *How does the CNN-based and ViT-based model performance compare across different tumor scales?*
   This research question focuses on evaluating our models across three different tumor scales including small-scale, medium-scale, and large-scale.

4. *What are appropriate strategies to optimize the performance of deep learning models for the segmentation of thorax CT / lung CT scans?*
   This research question addresses the appropriate optimization strategies of deep learning models for lung and lung cancer segmentation. The relevant strategies include experimenting with different model backbones, model architectures, and data preprocessing. The appropriateness of the strategies will be based on the overall performance improvement that includes Dice score and Hausdorff distance metrics.

1.2 Structure of the Thesis

This master thesis consists of six chapters, which are organized as follows:

Chapter 2 ***Background*:** introduces basics of Machine Learning and Deep Learning. Subsequently, it gives an overview of Convolutions Neural Network and Vision Transformers' key components. Furthermore, it explains the evaluation metrics used in this work. Finally, it provides a short description of Computed Tomography Imaging.

Chapter 3 ***Related Work*:** describes the current state-of-the-art related work for both semantic image segmentation and medical image segmentation tasks. Also, it focuses on general image segmentation (e.g. major organs) and lung cancer segmentation models.

Chapter 4 ***Methodology and Experimental Setup*:** explains the datasets, data-preprocessing, implementation of CNN-based and ViT-based models, experimental setup, and training details.

Chapter 5 ***Results*:** provides the results for CNN-based and ViT-based models based on lung segmentation and lung cancer segmentation datasets. In addition, it includes an in-depth comparison across different tumor scales for both CNN-based and ViT-based models.

Chapter 6 ***Discussion and Conclusion*:** reports key insights and contributions of the thesis. It also addresses limitations and provides an outlook on future work.

CHAPTER 2

Background

This chapter describes the relevant theoretical foundations for this thesis. Section 2.1 introduces Machine Learning fundamentals, including supervised learning and unsupervised learning. Subsequently, Section 2.2, explains Neural Networks and basic optimization algorithms. Convolutional Neural Networks and Vision Transformers are presented in Section 2.3 and Section 2.4, respectively. Subsequently, evaluation metrics used in this work are discussed in Section 2.5. Lastly, basic principles of Computed Tomography (CT) Imaging are provided in Section 2.6.

2.1 Machine Learning Fundamentals

Artificial Intelligence (AI) refers to technology capable of simulating human intelligence in tasks such as reading, speaking, and understanding the world. Machine Learning (ML), a subdiscipline of AI, aims to automatically map the input patterns (e.g. vehicle image) to the corresponding output values (e.g. truck). Traditional ML requires hand-crafting features which involve human expertise to select and prepare data for the model, it is often a time-consuming process. There are two major ML training paradigms, supervised (Section 2.1.1) and unsupervised (Section 2.1.2), each determined by the nature of the data and the specific objectives of the given task [Bis06].

2.1.1 Supervised Learning

Supervised learning involves training with $n \in \mathbb{R}$ labeled examples, where we have a training set $X_{train} = [x_1, \ldots, x_n]$ and a corresponding target vector $y_{train} = [y_1, \ldots, y_n]$. The predictive model is evaluated on $k \in \mathbb{R}$ new inputs from a test set $X_{test} = [x_1, \ldots, x_k]$, whereby the ultimate goal is to predict new, unseen labels of data [Bis06]. Supervised learning is divided into two major tasks: *classification* and *regression*. Discrete target values are predicted using the classification. The classification models can be used in tasks such as email spam detection (spam or not), while the regression models can be used for predicting continuous target values. An example of this is predicting housing prices based on the train data [Sim17].

2.1.2 Unsupervised Learning

In unsupervised learning, also known as self-supervised learning, the task of the ML model is to learn from unlabeled data. The training set does not include the corresponding target vector. Generally, the goal is to discover the underlying patterns within the data. The major tasks include *clustering* and *density estimation*. Clustering may be used to group similar data patterns, whereas density estimation determines the data distribution [Bis06]. For example, during pre-training, the model discovers the underlying patterns from unlabeled data and captures useful features.

2.2 Deep Learning

Deep Learning (DL) is a subset of the machine learning field. It is known to perform well in capturing complex features in high-dimensional data without the requirement of hand-crafting features [LBH15]. Therefore, it is often used in tasks such as object detection, speech recognition, and natural language processing. Its major disadvantage is a requirement for a large amount of training data.

2.2.1 Neural Networks

Neural Networks (NNs), also known as feedforward NNs, represent a foundational concept in the field of deep learning because they can in theory approximate any mathematical function, regardless of the complexity [Tur23a].

Consider a three-layer NN, which consists of the following layers: the input layer, the hidden layer, and the output layer. The input layer receives the raw input data as a vector $x \in \mathbb{R}$.

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \tag{2.1}$$

After that, the hidden layer performs a non-linear transformation on the data. This hidden layer can be denoted as follows:

$$\mathbf{h} = f(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \tag{2.2}$$

$$f(a) = \frac{1}{1 + \exp(-a)} \tag{2.3}$$

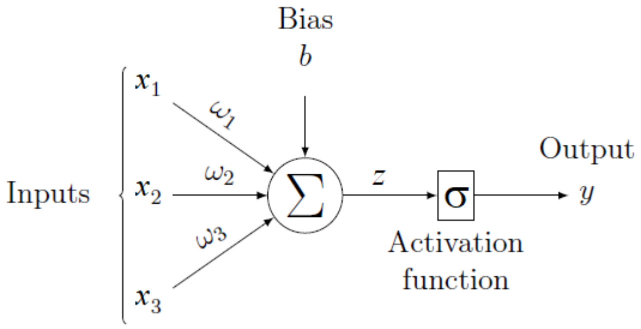


Figure 2.1: The Multi-layer Perception is the simplest form of Neural Network Architecture. It receives an input vector $\mathbf{x}$ to which it adds weights $\mathbf{W}^{(1)}$ and bias $\mathbf{b}^{(1)}$. Finally, it utilizes a non-linear activation function to compute output $\mathbf{y}$ [RHDN23].

The weight matrix $\mathbf{W}^{(1)}$ defines the relationships between input and first hidden layer using linear transformation, $\mathbf{b}^{(1)}$ represents the bias vector, and $f(a)$ is a non-linear activation function such as sigmoid in this case [Bis06] [Tur23b] [GBC16].

Finally, the output layer takes the output from the hidden layer and transforms it into the final output. The output layer $\mathbf{y}$ is denoted as follows:

$$\mathbf{y} = g(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)}) \tag{2.4}$$

where $\mathbf{W}^{(2)}$ defines the weights between the hidden layer and the output layer, $\mathbf{b}^{(2)}$ represents the bias for the output vector, and g function is an activation function that is determined depending on a task. This is also known as Multi-layer Perceptron (MLP) which is illustrated in Figure 2.1.

Generally, neural networks with multiple hidden layers are considered as *deep neural networks*, and they can capture more complex features in the deeper (hidden) layers [GBC16]. In the training process, each layer's parameters ($\mathbf{W}$, $\mathbf{b}$) are adjusted to minimize the error between the predictions $\hat{y}$ and the ground truth targets y. This is accomplished using the optimization methods such as gradient descent.

2.2.2 Optimization

During the training of the neural network, the key step is to adjust the parameters of the model to minimize the loss function $L(\hat{\mathbf{y}}, \mathbf{y})$, which estimates the cost between the predictions $\hat{\mathbf{y}}$ and actual targets $\mathbf{y}$. To achieve this, we can use one of the existing optimization algorithms such as Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent, Adaptive Moment Estimation (Adam), RMSProp, or Momentum [Rud17]. Backpropagation is crucial in neural network training, it involves a forward pass where inputs generate predictions and a backward pass where the gradient of the loss function is derived from the output layer back towards the input

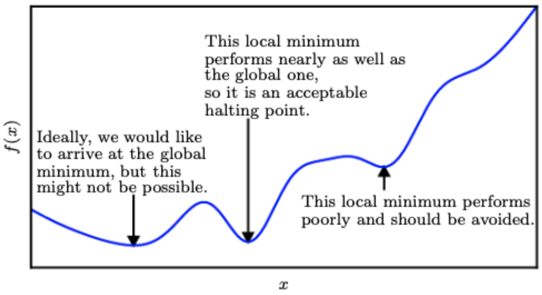


Figure 2.2: Gradient Descent Minimization Process. [GBC16]

layer [Bis06] [Ros21]. In the following, we provide a brief description of the commonly used optimization algorithms.

Batch Gradient Descent

The Batch Gradient Descent (BGD), also known as steepest descent, is the optimization method that minimizes the error function, $E(\mathbf{W}, \mathbf{b}) = E(\theta)$, for each parameter's update step on the whole training set [Bis06]. Iteratively, for each step r, the algorithm updates $\theta = (\mathbf{W}, \mathbf{b})$ towards the minimum, negative gradient $\nabla E(\theta)$:

$$\theta^{(r+1)} = \theta^{(r)} - \alpha \nabla E(\theta^{(r)}), \quad \text{where} \quad \alpha > 0. \tag{2.5}$$

α denotes the *learning rate*, which is used to determine the step size. Despite its simplicity, BGD is susceptible to converging on the local minimum rather than finding the global minimum (Figure 2.2).

For additional information, please refer to *Deep Learning* Chapter 4 [GBC16].

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) minimizes the error function $E(\theta)$ using a single randomly selected data point from the training set (x_i, y_i) for each update step. It can be denoted as:

$$\theta^{(r+1)} = \theta^{(r)} - \alpha \nabla E(\theta^{(r)}; (x_i, y_i)), \quad \text{where} \quad \alpha > 0. \tag{2.6}$$

When it comes to large datasets, BGD tends to be inefficient because it recomputes the gradients for the entire dataset before updating parameters. In contrast, SGD solves this by updating parameters after each data point, making it faster. Therefore, it usually converges faster than the BGD. However, SGD has a higher likelihood of missing the global minimum which makes it more unstable [Rud17].

Mini-Batch Gradient Descent

Mini-Batch Gradient Descent combines both BGD and SGD by utilizing a subset of p training examples for a single parameters update step [Rud17]. This approach can be represented as follows:

$$\theta^{(r+1)} = \theta^{(r)} - \alpha \nabla E(\theta^{(r)}; (x_{i:i+p}, y_{i:i+p})), \quad \text{where} \quad \alpha > 0. \tag{2.7}$$

The notation i stands for the starting position, and $i+p$ defines the final position of a subset from a training set.

Further details are available in *Deep Learning* Chapter 8 [GBC16].

Adaptive Moment Estimation

Adaptive Moment Estimation (Adam) combines both AdaGrad and RMSProp optimizers [KB17]. AdaGrad adapts the learning rate according to the frequency of parameter updates; more frequent means smaller updates and vice versa. On the other hand, RMSProp uses an adaptive learning rate method that is similar to Adadelta which aims to reduce the learning rate [Rud17].

2.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks, also known as CNNs, is a widely spread deep learning architecture used in various ML tasks such as computer vision, natural language processing, and speech recognition. LeCun et al. [LBD+89] introduced it in 1989 to classify handwritten zip code digits. This section describes the basic components of CNNs, which are visualized in Figure 2.3 [GWK+15].

2.3.1 Convolutional Layer

The convolutional layer uses an input image I of dimension (h x w x d), with an argument kernel K of dimension (k_h x k_w x d) , also known as filter, to generate a feature map ($I * K$) with a dimension of $(h - k_h + 1)$ x $(w - k_w + 1)$ [Rag18]. The I height, width, and depth are denoted as h, w, and d, respectively. Essentially, the kernel matrix is sliding over the input image matrix, multiplying each corresponding element. Subsequently, an output matrix is summed and assigned to the feature map (Figure 2.4). Goodfellow el at. [GBC16] have denoted this for two-dimensional I and two-dimensional K as follows:

$$C(i,j) = (I * K)(i,j) = \sum_h \sum_w I(h,w)K(i-h, j-w) \tag{2.8}$$

where i and j denote a position within I [GWK+15] [Rag18].

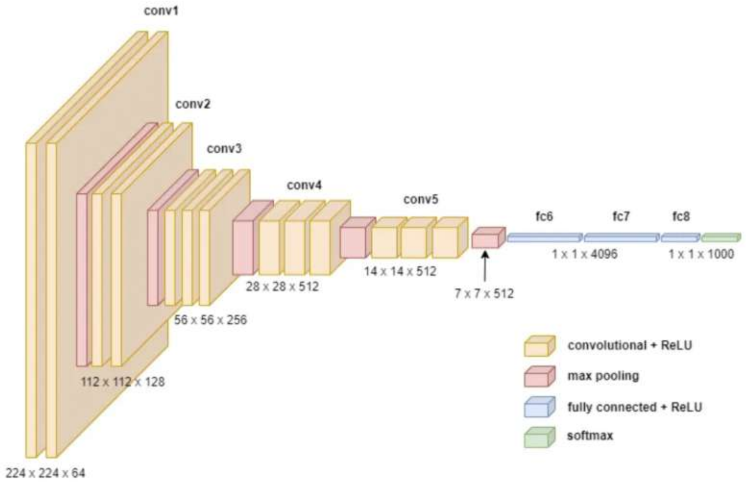
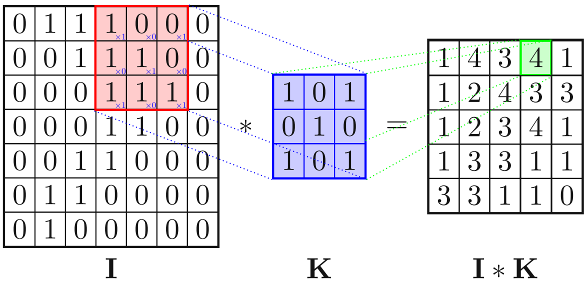


Figure 2.3: Convolutional Neural Network consists of multiple convolution layers, non-linear operations, and pooling layers. After the FC layer, softmax is used to determine a predicted class [GBC16] [VGG24].



$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0&1&1&1&0&0&0\\\hline 0&0&1&1&1&0&0\\\hline 0&0&0&1&1&1&0\\\hline 0&0&0&1&1&0&0\\\hline 0&0&1&1&0&0&0\\\hline 0&1&1&0&0&0&0\\\hline 0&1&0&0&0&0&0\\\hline\end{array} * \begin{array}{|c|c|c|}\hline 1&0&1\\\hline 0&1&0\\\hline 1&0&1\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 1&4&3&4&1\\\hline 1&2&4&3&3\\\hline 1&2&3&4&1\\\hline 1&3&3&1&1\\\hline 3&3&1&1&0\\\hline\end{array}$$

Figure 2.4: Kernel matrix K with dimensions 3x3 sliding over an input image I with dimensions 7x7 performing convolution. The convolution operation is denoted with an asterisk. [Exc19]

2.3.2 Non-Linear Operations

Non-linear operations or activation functions are applied to feature maps to capture nonlinear relationships. For example, Rectified Linear Unit (ReLu) is an important activation function which can be mathematically expressed as follows:

$$f(x) = max(0, x) \tag{2.9}$$

Also, other common non-linear activation functions include leaky ReLu hyperbolic tangent function (Tanh), Sigmoid, Binary step function, and Softmax. There is no strict rule when selecting an activation function since it depends on the task. However, ReLu is generally accepted as a starting point [GBC16].

2.3.3 Pooling Layer

The pooling layer is generally applied after a convolutional layer to reduce specific input dimensions using operations such as max pooling, average pooling, or sum pooling which significantly reduces computational costs. In the most commonly used max pooling operation, the largest element from each patch of input data is assigned to a feature map. However, this might lead to a certain information loss [GBC16] [Nan23].

2.3.4 Fully Connected Layer

After extracting the input features, the feature map matrices are flattened into a one-dimensional vector $\mathbf{x}$ which is used as an input for the fully connected (FC) layer. In the FC layer, each neuron is connected to all neurons of the previous layer. Finally, an output layer $\mathbf{y}$ uses the activation function $\mathbf{g}$ (e.g. Softmax) to classify outputs into a corresponding class [GBC16].

2.3.5 Skip Connections

Skip connections, as their name suggests, allow for skipping one or multiple layers in the NN. There are deeper CNN-based architectures such as ResNet that include skip connections to avoid vanishing gradient problems [HZRS15]. During backpropagation, gradients are used to update an NN's weights, however, if gradient error becomes very small in early layers, it leads to a vanishing gradient problem. Therefore, skip connections are used to tackle this problem by allowing gradients to bypass multiple layers.

2.4 Vision Transformers (ViT)

In 2017, Vaswani et al. [VSP+23] introduced a new network architecture namely Transformer (Figure 2.5). The transformer architecture achieved impressive results in the Natural Language Processing (NLP) field [DBK+21]. However, CNN-based architectures remained the gold standard when it comes to computer vision tasks. In 2021, Dosovitskiy et al. [DBK+21] introduced a Vision Transformer based on Self-Attention architecture. The aim was to extend Transformers to ViT models used for computer vision tasks. On many image classification tasks, ViT demonstrated the same or better results compared to state-of-the-art CNNs with fewer computational resources [DBK+21].

2.4.1 Self-Attention

The transformer is based entirely on attention mechanisms, which allows the model to focus on specific parts of the given input rather than treating each part equally like CNNs. This is based on single-headed attention, also known as "Scaled-Dot-Product Attention", which is used to compute an attention vector for each input element. It consists of query Q, key K, and value V vectors that extract different components from every input element. Q represents certain information that we want to extract from data,

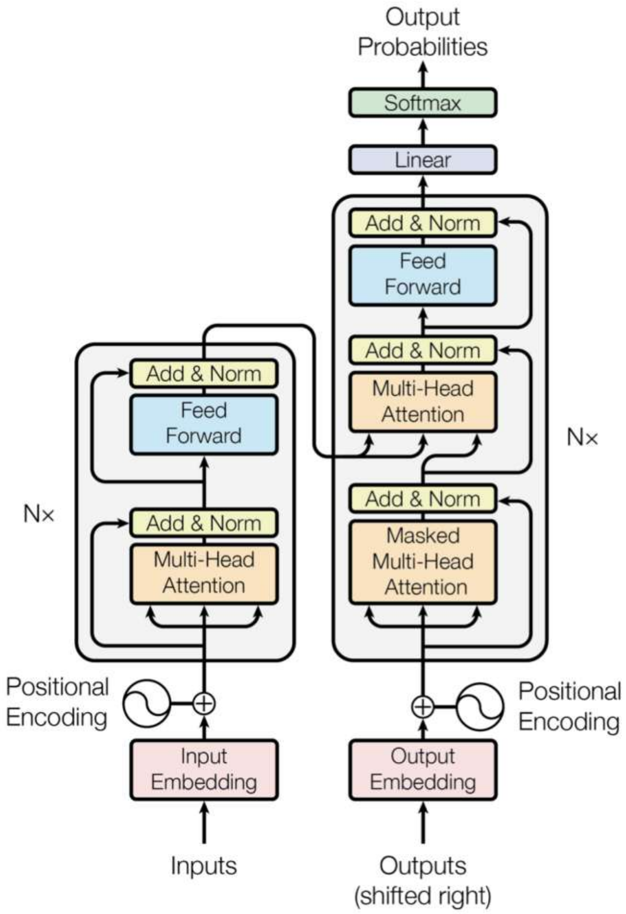


Figure 2.5: Transformer Architecture Visualization. The left side represents an encoder block that consists of positional encoding, a Multi-Head attention layer, and a feed-forward layer. The right side shows a decoder block, which includes the Masked Multi-Head Attention layer, Multi-Head attention layer, and Feed Forward layer. Finally, it uses a Linear layer and Softmax to output the next-token probabilities. [VSP+23]

K is defined as a summary or index for each element, and V contains all information about the element. This is given as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.10}$$

where $\sqrt{d_k}$ represents a key dimensions.

Moreover, this paper introduces a beneficial "Multi-Head Attention" mechanism that allows to input multiple elements Q, K, V as multiple weight matrices QW_i^Q, KW_i^K and VW_i^V. The Multi-Head computation with h parallel attention layers is defined by:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.11}$$

where weight matrices are defined as $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$. Intuitively, the d_v and d_{model} stands for value and model dimensions. Multi-Head attention creates multiple attention vectors, enabling efficient (parallel) context analysis for each word. Unlike Single-headed attention, it uses multi "heads" to capture a wider range of context and relationships within the data. Lastly, these attention vectors are concatenated into a single-headed attention vector which is fed into NN.

2.4.2 ViT Architecture

An overview of the vision transformer architecture is illustrated in Figure 2.6. The input image is first partitioned into patches (Figure 2.6, bottom left). These fixed-size patches use positional encoding because it allows for capturing sequence order to ensure the context of an image. This sequence is normalized for each block using Layernorm (LN), which stabilizes the training process [WLX$^+$19]. After that, the transformer model applies the previously explained Multi-Head attention mechanism and a feedforward NN (Figure 2.6, right). Finally, the MLP head is used now for classifying an object into a corresponding class such as a bird vs car (Figure 2.6, top left).

2.5 Evaluation Metrics

In the computer vision field, the Dice Similarity Coefficient and Hausdorff distance are prominent evaluation metrics, especially in the segmentation tasks. On the one hand, the Dice Similarity Coefficient measures the overlap between predicted and ground-truth segmentation (Section 2.5.1). On the other hand, the Hausdorff distance offers a geometric interpretation by measuring the dissimilarity between two given finite point sets (Section 2.5.2). These metrics can be used for both two-dimensional and volumetric analysis [ZWB$^+$04] [HKR93].

2.5.1 Dice Similarity Coefficient

Dice Similarity Coefficient (DSC), also known as the F1-score or Dice score, is a statistical metric used to quantify the similarity between two binary vectors (e.g. image). It can be also defined as a harmonic mean of precision and recall. Precision is a percentage of relevant over all retrieved predictions, while recall uses only relevant predictions [Pre24]. Its primary application is to measure the accuracy of a predicted image segmentation against the ground truth segmentation [ZWB$^+$04] [Dic45]. It is defined as follows:

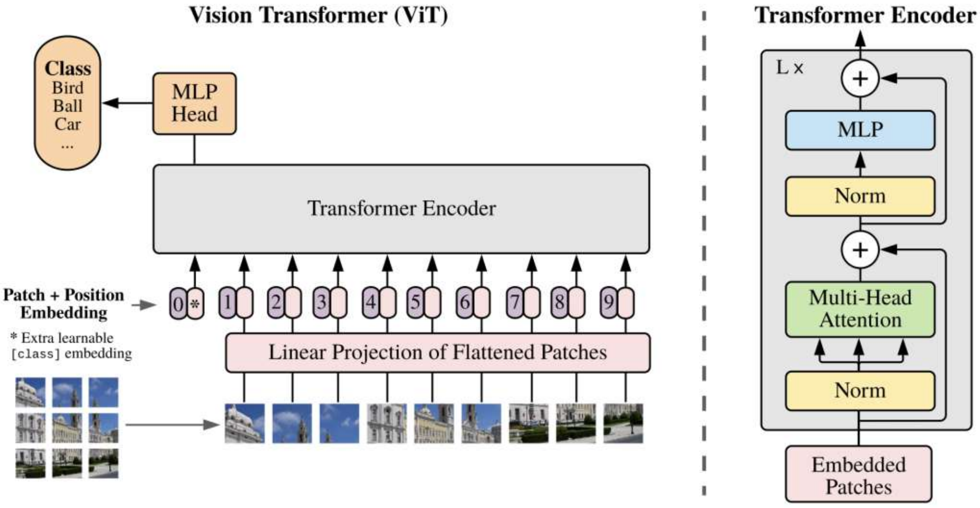


Figure 2.6: Vision Transformer Architecture (left) and Transformer Encoder Visualization (right). ViT partitions an image into patches with positional encoding. These patches are flattened into a one-dimensional sequence that is fed into the transformer encoder. The right side visualizes a transformer encoder in detail. Finally, the MLP head outputs the corresponding class [DBK+21].

$$DSC = \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}|}. \tag{2.12}$$

y represents ground truth segmentation and $\hat{y}$ the predicted segmentation. The DSC can range between 0 and 1. If $DSC = 0$, there is no overlap; $0 < DSC < 1$ relates to partial overlap; $DSC = 1$ represents perfect overlap between y and $\hat{y}$, as visualized in Figure 2.7 [ZWB+04]. DSC can be defined in terms of true positives (TP), true negatives (TN), and false negatives (FN) as follows:

$$DSC = \frac{2TP}{2TP + FP + FN}. \tag{2.13}$$

These terms can be expressed in the context of two-dimensional image pixels:

- True Positive (TP): Positive pixel correctly classified.
- True Negative (TN): Negative pixel correctly classified.
- False Positive (FP): Positive pixel incorrectly classified.
- False Negative (FN): False pixel incorrectly classified

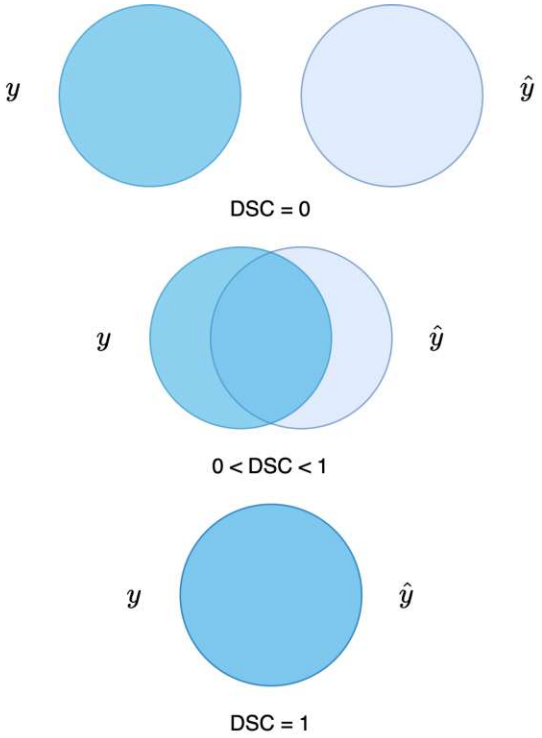


Figure 2.7: This figure provides a graphical representation of the Dice Similarity Coefficient (DSC) metric, which is often used in medical image segmentation. A $DSC = 0$ implies no overlap, $0 < DSC < 1$ includes partial overlap, and $DSC = 1$ indicates perfect overlap of y and $\hat{y}$.

DSC metric is crucial for the evaluation of both two-dimensional and three-dimensional images. In addition, it is the most widely adopted metric in evaluating medical image volume segmentation [TH15b].

2.5.2 Hausdorff Distance

The Hausdorff Distance (HD) is a commonly used metric in computer vision tasks, that measures the distance between two finite point sets [TH15a]. The directed HD is given by:

$$h(U, V) = \max_{u \in U} \min_{v \in V} ||u - v|| \tag{2.14}$$

where $||u - v||$ is a norm such as L2 or Euclidean distance. It takes the point within the set U with the maximum distance from any point in V, and computes the distance from this point $u \in U$ to the nearest point $v \in V$ [HKR93]. The HD for two point sets, $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$, is given by:

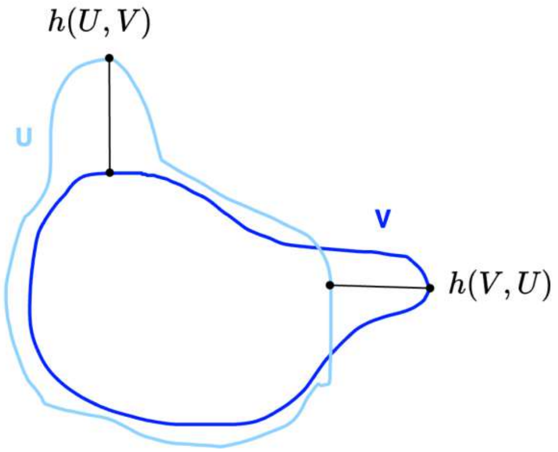


Figure 2.8: This figure visualizes Hausdorff Distance (HD) that measures the distance between two finite point sets [TH15a].

$$H(U, V) = max(h(U, V), h(V, U)). \tag{2.15}$$

Finally, the $H(U, V)$ takes the maximum of values $h(U, V)$ and $h(V, U)$. Therefore, it measures the dissimilarity between two sets by measuring the maximum distance from a point $u \in U$ to any point in set V and the reverse [HKR93]. This is visualized in Figure 2.8.

2.6 Computed Tomography (CT) Imaging

Computed Tomography (CT) Imaging uses a series of patient X-ray observations from different angles to capture cross-sectional images or slices (Figure 2.9). These slices are stacked as a 3D volume image which can show the patient's internal structures such as organs, tissues, or lesions. Depending on the CT machine, the tissue thickness ranges between 1-10 millimeters. Slices or tomographic images contain more information compared to traditional X-rays [CT224]. CT scans are expressed with Hounsfield units (HU):

$$HU = \left(\frac{\mu_{\text{material}} - \mu_{\text{water}}}{\mu_{\text{water}}} \right) \times 1000 \tag{2.16}$$

where μ represents CT linear attenuation coefficient. The attenuation coefficient measures the amount of lost energy when a narrow beam of X-rays passes through the material [CT224] [Hou]. Finally, CT scans are useful for detecting various abnormalities such as lung cancer which is one of the deadliest cancers [BFS$^+$18].

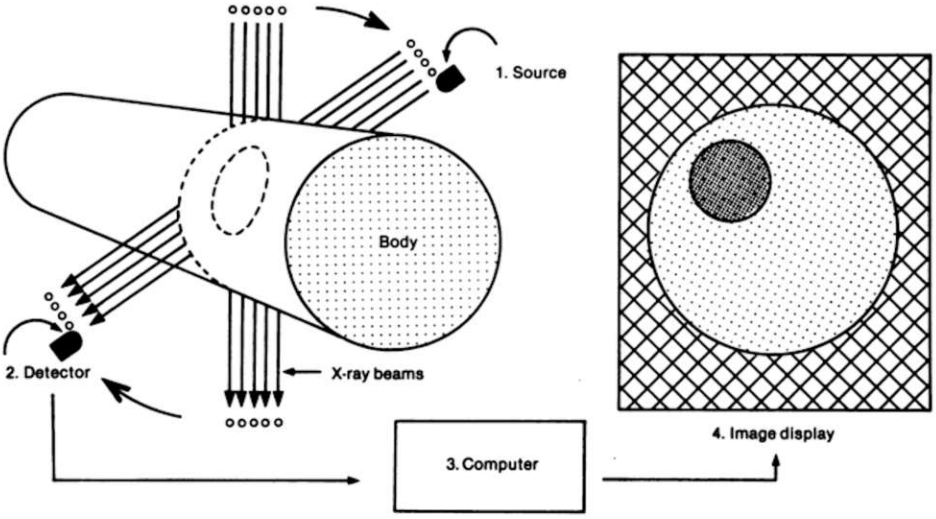


Figure 2.9: Computed Tomography (CT) Scanner Schematic Overview. The source uses X-rays that pass through the body and the detector captures information. Finally, computer is used to reconstruct and display images [CT224] [sch24].

CHAPTER 3

Related Work

This chapter describes the current state-of-art solutions for image segmentation, progressing from general medical segmentation (e.g. major organs) to more specific segmentation tasks. In Section 3.1, we address the semantic image segmentation state-of-the-art research. Section 3.2 describes image segmentation in the field of medicine. We list the most relevant research for image segmentation tasks for both CNN-based and ViT-based models. Finally, we provide an overview of state-of-the-art approaches concerning lung cancer segmentation.

3.1 Semantic Image Segmentation

Semantic image segmentation refers to classifying each pixel of a given image into a certain class. Humans can perform many of these tasks that involve both detecting and segmenting objects with relatively little cognitive effort. Interestingly, humans are even able to perform an image segmentation with unknown objects, for example, objects within medical X-ray scans, however, these tasks are usually complex, costly, and time-consuming ([GLGL17], [Jai20]). While the main focus of this work is on medical image segmentation tasks, this section provides a brief description of semantic image segmentation work in a broader sense for both CNN-based and ViT-based architectures.

3.1.1 CNN-based Segmentation Models

Generally, Convolutional Neural Networks can automatically extract high-level features (e.g. objects) and low-level features (e.g. colors) from an image [Jai20]. This has led to the development of many CNN architectures for semantic image segmentation such as SegNet [BKC16], U-Net [RFB15], DeepLabv3 [CPSA17], DeepLabv3+ [CZP+18], PSPNet [ZSQ+17], PAN [LXAW18], FPN [LDG+16], and LinkNet [CC17].

SegNet uses encoder-decoder architecture for pixel-wise semantic segmentation [BKC16]. Its encoder network with 13 convolutional layers is similar to the VGG16 network [SZ15]. The decoder uses max-pooling to optimize efficiency in terms of memory, however, this loss of information may lead to worse performance in fine-grained segmentation [BKC16]. The U-Net is prominent encoder-decoder architecture which was initially designed for medical image segmentation, it uses skip connections which improve capturing of semantic features (Subsection 3.2.1).

DeepLabv3, an improvement from previous DeepLab versions, uses atrous convolution for semantic image segmentation that captures multi-scale context [CPSA17]. DeepLabv3+ combines atrous convolution with encoder-decoder architecture which increases the segmentation quality [CZP+18]. The major disadvantage of DeepLabv3 and DeepLabv3+ is increased computational complexity.

PSPNet is based on a pyramid pooling module for semantic segmentation that effectively captures global context information. It achieves state-of-the-art results when it comes to scene parsing [ZSQ+17]. Scene parsing divides an image into multiple segmentation regions that belong to a certain semantic category such as a building, human, or wall. Also, previous work applies 3D PSPNet on multi-scale global contextual semantic segmentation tasks ([FL19], [Wan20]). PAN expands this by combining attention mechanism and pyramid pooling [LXAW18]. Both of these methods suffer from high computational costs.

There are two excellent methods in similar computer vision tasks, especially object detection ([GLL19], [WGC+20], [XYZ+19], [LWT+21]). Feature Pyramid Network (FPN) architecture, which significantly improves feature extraction in multi-scale detection tasks [LDG+16], and LinkNet that efficiently links the encoder with the decoder [CC17].

When it comes to image classification, some of the most prominent CNN backbones include ResNet [HZRS15], DenseNet [HLvdMW18], and EfficientNet [TL19]. These backbones can be also utilized for semantic segmentation tasks since they help extract useful features.

In 2015, He et al. [HZRS15] demonstrated extended residual nets with up to 152 layers with relatively low complexity. These achieved first place in several tasks that include ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation as a part of ILSVRC & COCO 2015 competitions.

DenseNet, introduced by Huang et al. [HLvdMW18] in 2018, is a network architecture that connects each layer to every other layer in a feed-forward manner. It was tested on CIFAR-10, CIFAR-100, SVHN, and ImageNet object recognition tasks, where it demonstrated substantial improvements over the state-of-the-art at that time while requiring fewer computing resources.

In 2019, Tan et al. [TL19] achieved first-place accuracy on ImageNet, with their EfficientNet model being 8.4x smaller and 6.1 faster than the previously leading model.

3.1.2 ViT-based Segmentation Models

In 2017, Vaswani et al. [VSP+23] proposed Transformers, which were initially used for machine learning translation with state-of-the-art performance in various NLP tasks [DBK+21]. Generally, large transformer models are pre-trained on large text corpora and then fine-tuned on a specific task, such as the language representation model BERT [DCLT19]. In 2021, Dosovitskiy et al. [DBK+21] introduced the Vision Transformer Model for classification tasks that apply a transformer model architecture directly to images. This is done by using a sequence of image patches instead of word tokens (Section 2.4.2). It achieves a similar performance compared to similar-sized EfficientNet and ResNet models [DBK+21].

Various ViT-based models achieve excellent results on semantic segmentation tasks such as Segmenter [SGLS21], SegViT [ZTT+22], Swin Transformer [LLC+21], BEiT [BDPW22], SERT [ZLZ+21], and Trans2Seg [XWW+21].

Segmenter utilizes pre-trained models for image classification, but it is fine-tuned on semantic segmentation datasets. Its performance significantly drops with smaller datasets which can be a major limitation [SGLS21]. SegViT introduces the Attention-to-Mask mechanism that enriches contextual information and improves computational efficiency [ZTT+22]. Also, it outperforms models that rely on plain ViT backbone. However, the Attention-to-Mask mechanism requires a large amount of GPU memory that may not be supported. Swin Transformer employs a hierarchical architecture with a shifted window scheme, which improves efficiency in a wide range of computer vision tasks such as semantic segmentation and image classification [LLC+21].

Inspired by BERT [DCLT19] that was designed for natural language processing, BEiT is a self-supervised vision model. It creates visual tokens from an image that are fed into the ViT backbone without requiring labeled data [BDPW22]. Nonetheless, it may not be able to capture certain image elements that are relevant to the task. SERT adopts a sequence-to-sequence perspective to improve the relationships in semantic segmentation [ZLZ+21]. Trans2Seg demonstrates an advantage over CNN architectures in terms of global image context, although, it may lead to worse performance in fine-grained segmentation [XWW+21].

In this work, we have utilized the ViT-based Segment Anything Model (SAM) introduced by Kirillov et al. [KMR+23] in 2023. This model was trained on the world's largest segmentation dataset (until this date) which includes over 1 billion masks. The SAM concept is slightly different compared to traditional ML models because it is designed to be promptable. For instance, humans can create a bounding box, point, or text prompt. This allows for excellent zero-shot generalization capabilities on unseen data, in some cases even competitive with prior state-of-art fine-tuned models. However, it requires human input during inference (Section 4.5) [BMR+20][KMR+23].

3.2 State-of-the Art: Medical Image Segmentation

In this section, we describe general medical image segmentation, particularly larger anatomical structures such as organs. This is motivated by our task of lung segmentation which is a larger target compared to lung cancer segmentation. Lastly, we provide lung cancer segmentation related work.

3.2.1 Medical Image Segmentation

CNNs are often used when it comes to medical image segmentation such as bones, blood vessels, and major organs [KJvdS17]. There are many CNN-based architecture variants such as U-Net [RFB15] or U-Net++ [ZSTL18] that are specifically designed for medical image segmentation tasks.

We have used the lung cancer dataset from the Medical Segmentation Decathlon (MSD) [ARB+22], which includes various biomedical segmentation challenges to find the best model in terms of generalizability. The MSD consists of ten different tasks such as brain, lung, and pancreas segmentation. Currently, the best-performing model across all ten MSD tasks is the Swin UNETR model with 76.68% average dice score [TYL+22]. It uses a Transformer-based U-shaped encoder-decoder architecture that improves global context in semantic segmentation. Also, Swin UNETR is a 3D VIT-based model pre-trained on 5,050 CT scans from various organs. There are other leading models on the MSD leaderboard such as DiNTS [HYR+21], nnUNet [IPK+18], and Model Genesis [ZSS+19].

DiNTS focuses on optimal network topology for 3D medical image segmentation tasks, while minimizing GPU memory usage [HYR+21]. nnUNet employs a self-adapting framework that can generalize different 3D medical image segmentation tasks [IPK+18]. Model Genesis is based on self-supervised learning for pre-training, followed by fine-tuning on medical image segmentation tasks [ZSS+19]. However, this pre-training phase may be computationally heavy.

For lung segmentation in particular, Skourt et al. [ASEHM18] employed U-Net architecture that achieved a dice score of 0.9502 on their manually segmented dataset with a few hundred lung CT scans. Similarly, Jalali et al. [JFR+21] utilized U-Net for lung CT segmentation, however, they replaced the encoder with ResNet34 network [HZRS15]. Their results surpassed the previously mentioned publication. There are many similar implementations available for lung segmentation ([HPP+20], [NTBA22], [HCP+23], [AHL+20], [GML20]). The majority of lung segmentation datasets include X-ray images ([WYB+23], [LLY+22], [Che24]).

There are many available publications that compare CNN-based models and Transformers ([HEO22], [MHSS21], [DSY+22], [SFW+23]). Jia et al. [JBZ+22] explore the efficiency between U-Net and Transformer-based models in the field of medical imaging, using two public 3D brain datasets. The results indicate that vanilla U-Net can outperform the Transformer-based model with only a slight modification. Their research demonstrates that U-Net is still a highly competitive architecture in medical imaging.

When it comes to combining CNN-based models and Transformers, research conducted by Cao et al. [CWC+23] proposes Swin-Unet that is based on Transformer U-shaped Encoder-Decoder architecture with skip connections. Swin-Unet outperformed both CNN-based models and combinations of Transformer and CNN-based models for tasks on multi-organ and cardiac image segmentation. Lan et al. [LCJ+24], propose a BRAU-Net++ CNN-Transformer network that improves learning of long-range dependency using self-attention. PMTrans is a Pyramid Medical Transformer that leverages multi-scale attention and CNN-based feature extraction [ZZ22]. Similarly to BRAU-Net++, it can efficiently capture long-range dependencies.

ViT-based SAM image encoder relies on masked autoencoder which masks certain input patches and restores missing pixels [HCX+21]. A SAM fine-tuned for medical images was implemented by Huang et al. [HYL+24]. They applied SAM on a large COSMOS 1050K dataset that includes various medical objects such as eyeballs, optic nerve, lips, liver, hip, femur, spleen, and kidney. Fine-tuned SAM for medical images showed impressive performance in certain anatomical structures such as the humerus, however, it performed poorly in cases with lower contrast or weak boundaries [ZSJ24]. Also, SAM with ViT-Huge backbone demonstrated significantly better performance compared to ViT-Base backbone (smaller version) according to Huang et al. [HYL+24]. Many similar implementations of SAM-based models have been evaluated on various medical imaging datasets ([MDG+23a], [HBL+23], [ZL23], [WJL+23]). Interestingly, He et al. [HBL+23] report that SAM performs significantly worse compared to five state-of-the-art algorithms for medical image segmentation. Ma et al. [MHL+23] introduced the MedSAM project that fine-tunes the Segment Anything Model on medical images for binary segmentation.

3.2.2 Lung Cancer Segmentation

Numerous publications use deep learning for lung cancer segmentation ([CWP+19], [JHS+21], [LDD+18], [WZL+17], [PIVT+22a], [ARB+22], [IPK+18], [PPKS23], [TT23], [FFC+23]).

Chen et al. [CWP+19] utilize a hybrid segmentation network, based on 2D and 3D CNNs, which is used for small-cell lung cancer segmentation (SCLC). They use Dice loss to improve their results on highly imbalanced datasets. Also, 3D CNNs were used for both detection and segmentation of NSCLC that spreads towards the brain [JHS+21]. Liu et al. [LDD+18] tackle a problem of lower-quality CT scans and lack of annotated data by utilizing an object detection neural network for lung nodule segmentation. Central Focused CNNs can accurately segment lung nodules from different CT scan datasets with minimal difference [WZL+17].

This thesis was inspired by the Primakov et al. [PIVT+22a] workflow, where they first isolate lungs and subsequently segment the NSCLC (Figure 3.1). They report that 56% of radiologists prefer automatic segmentation over manual segmentation. In their approach, a three-step workflow is proposed. In the first step, image preprocessing is conducted for each dataset (1414 NSCLC patients), the second step focuses on lung isolation, and finally, a 2D U-Net is employed to detect and segment tumors. To improve the model's performance, they replaced ReLU activations with Exponential Linear Unit (ELU). The loss function was defined with combined Dice Similarity Coefficient (DSC) loss and binary cross-entropy loss.

Furthermore, the MSD [ARB+22] NSCLC dataset, which was utilized in this work, consists of 3D volumes CT modality for 96 patients. Isensee et al. [IPK+18], introduced no-new-Net (nnU-Net) that is a self-adapting framework based on 2D and 3D vanilla U-Nets [PPKS23]. At the time of the MSD challenge, nn-U-Net achieved the best performance on the phase 1 MSD leaderboard. In addition, nn-U-Net had the highest mean dice score of 69% on the NSCLC dataset.

A vast majority of lung cancer segmentation methods rely on CNN-based architectures, but some utilize ViT-based models [TT23]. Fanizzi et al. [FFC+23] implemented multiple transformer architectures, including pre-trained ViT, Pyramid ViT, and Swin Transformer for NSCLC segmentation ([DBK+21], [WXL+21], [LLC+21]). Nevertheless, these attempts did not show any improvement compared to traditional state-of-the-art CNNs.

Despite this progress in the field of lung cancer segmentation, there remains an opportunity to further explore the difference between CNN-based and SAM-based models, especially across different tumor sizes. This work aims to implement and compare CNN-based models with different backbones, SAM with simulated human interaction, and SAM without simulated human interaction on lung segmentation and lung cancer segmentation tasks. Finally, we evaluate the performance on test sets using the Dice Similarity Coefficient and Hausdorff Distance metrics.

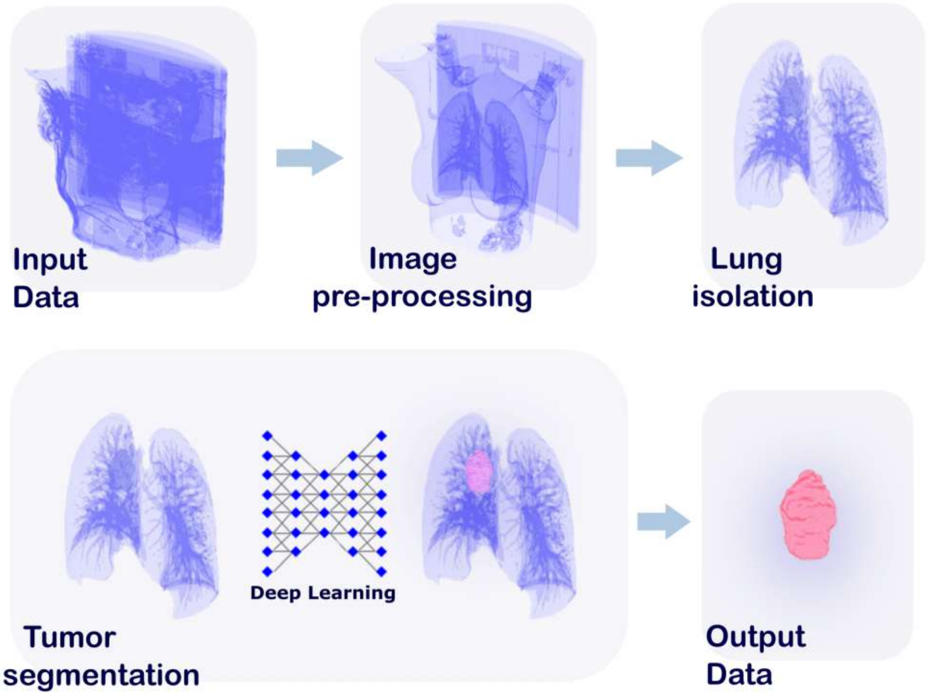


Figure 3.1: Key steps for fully-automatic non-small cell lung cancer segmentation workflow proposed by Primakov et al. [PIVT$^+$22a]. It combines both lung isolation and tumor segmentation tasks.

CHAPTER 4

Methodology and Experimental Setup

This chapter describes the data characteristics, data-specific and model-specific data preprocessing, model design, hyperparameter tuning, model selection, and evaluation. The overall process overview is given in Section 4.1. The lung segmentation dataset D_1 and lung cancer segmentation D_2 are explained in Section 4.2. We explain used preprocessing and data augmentation techniques in Section 4.3. After that, we describe CNN-based models and SAM-based models implementation in Section 4.4 and Section 4.5, respectively. Hyperparameter tuning and model selection are defined in Section 4.6. Section 4.7 outlines the experimental setup. We describe the training details of our all models and experiments in Section 4.8.

4.1 Process Overview

In this section, we present the process overview of the thesis (Figure 4.1). In the first step, we started with data acquisition. After that, we conducted data preprocessing that varies based on data complexity. In the model development part, we re-implemented CNN-based and SAM-based models with various backbones and model architectures. In the last step, we evaluated our models including CNN-based models, SAM with simulated human interaction (SHI), and SAM without SHI. In addition, we evaluated these models for three different tumor scales including small, medium, and large.

1. **Data Acquisition**
   This phase required collecting data from various publicly available datasets. We used the lung segmentation dataset D_1 to include a dataset with a larger target. Regarding the lung cancer dataset, we ended up using the Medical Segmentation Decathlon lung cancer segmentation dataset D_2 [ARB$^+$22] because there was

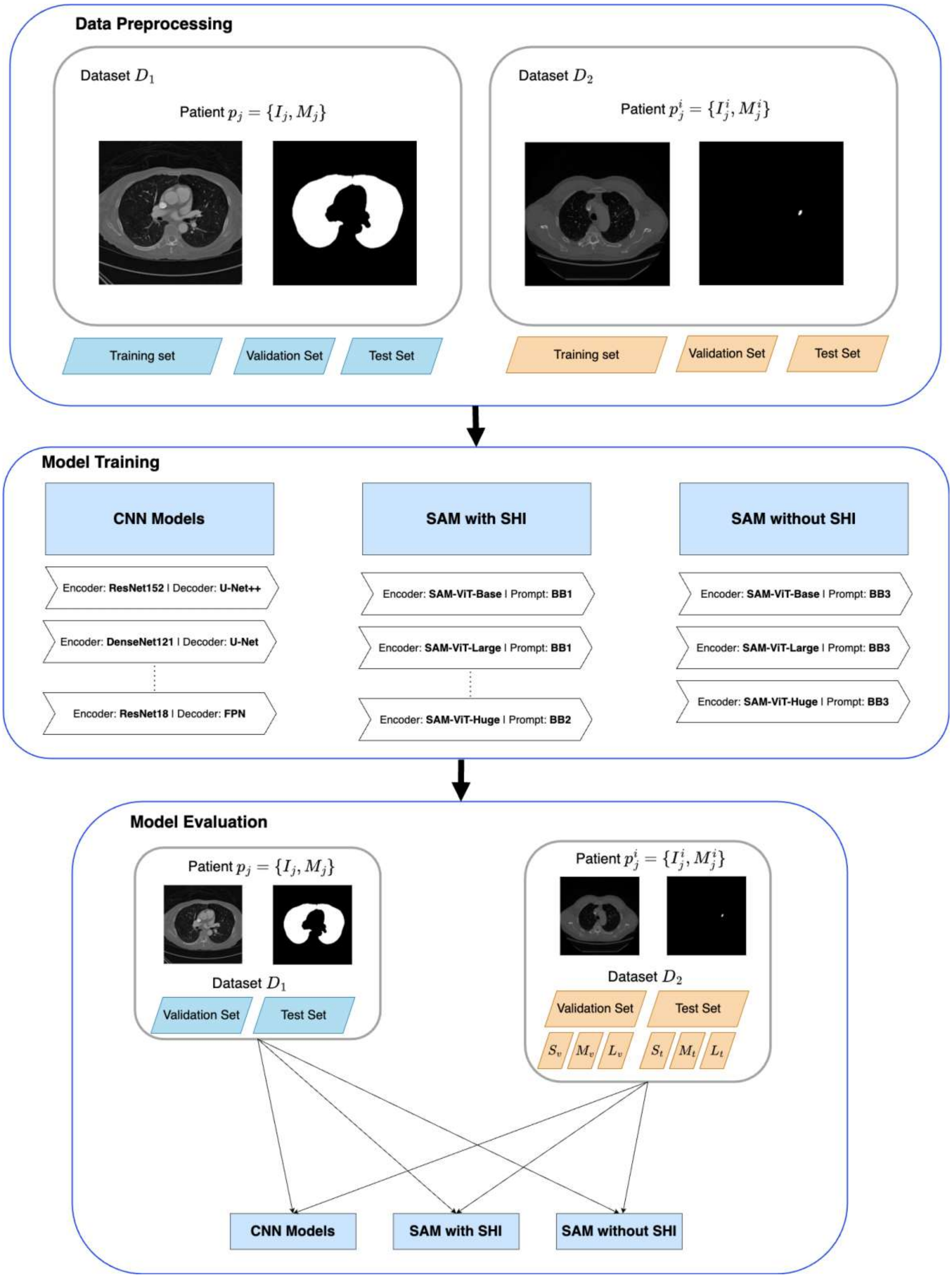

Data Preprocessing
Dataset D_1
Patient $p_j = \{I_j, M_j\}$
Dataset D_2
Patient $p_j^i = \{I_j^i, M_j^i\}$
Training set
Validation Set
Test Set
Training set
Validation Set
Test Set
Model Training
CNN Models
SAM with SHI
SAM without SHI
Encoder: ResNet152 | Decoder: U-Net++
Encoder: DenseNet121 | Decoder: U-Net
Encoder: ResNet18 | Decoder: FPN
Encoder: SAM-ViT-Base | Prompt: BB1
Encoder: SAM-ViT-Large | Prompt: BB1
Encoder: SAM-ViT-Huge | Prompt: BB2
Encoder: SAM-ViT-Base | Prompt: BB3
Encoder: SAM-ViT-Large | Prompt: BB3
Encoder: SAM-ViT-Huge | Prompt: BB3
Model Evaluation
Patient $p_j = \{I_j, M_j\}$
Dataset D_1
Validation Set
Test Set
Patient $p_j^i = \{I_j^i, M_j^i\}$
Dataset D_2
Validation Set
Test Set
S_v
M_v
L_v
S_t
M_t
L_t
CNN Models
SAM with SHI
SAM without SHI

Figure 4.1: Key Steps of Process Overview. It starts with data acquisition, subsequently, we perform data preprocessing that splits the data into training, validation, and test sets for both dataset D_1 and D_2. After that, we train CNN-based, SAM with SHI, and SAM without SHI models on the training set, and evaluate on validation and test sets. Bounding boxes are denoted as BB. In addition, we evaluate all models on small-scale, medium-scale, and large-scale tumors.

a substantial amount of existing literature and implementations available. An in-depth description of these datasets is given in Section 4.2.

2. **Data Preprocessing**
   The same data preprocessing was applied to training, validation, and test sets for each dataset. For the lung segmentation dataset D_1, no preliminary preprocessing was required since it consists of central CT slices. In contrast, the lung cancer segmentation dataset D_2 included non-tumor slices from 3D CT scans which were filtered to include only 2D positive CT slices to ensure compatibility with SAM, see Section 4.3.

3. **Model Design**

   - **CNN-based Models**
   - **ViT-based Models**
     - **SAM with Simulated Human Interaction**
     - **SAM without Simulated Human Interaction**

   In this step, we implemented two model architectures, CNN (Section 4.4) and ViT (Section 4.5). For CNN-based models, we used different network encoders (backbones) and model architectures, such as ResNet152 with U-Net. We used the Segment Anything Model with SHI and without SHI. In addition, we used three backbones for both SAMs including SAM-ViT-Base, SAM-ViT-Large, and SAM-ViT-Huge. We have trained and evaluated these models on both tasks: lung segmentation dataset D_1 and lung cancer segmentation dataset D_2. The focus of this thesis was on the lung cancer segmentation dataset D_2 due to its highly complex region of interest (ROI). All experiments were conducted on the high-performance cluster (HPC) of the CIR lab.

4. **Hyperparameter Tuning and Model Selection**
   In Section 4.6, we specify the configurations used for our experiments to ensure reproducibility. This includes hyperparameters for both CNN-based models and ViT-based models. CNN-based model hyperparameters consist of model architecture, network encoder, optimizer, loss function, and number of training epochs. In contrast, ViT-based model SAM hyperparameters include backbones, bounding box prompts, and number of training epochs.

5. **Evaluation**
   A validation set is used for hyperparameter tuning to optimize model performance. Dice Similarity Coefficient (DSC) and Hausdorff Distance (HD) metrics were employed to evaluate model generalization performance on the test sets for both dataset D_1 and dataset D_2 (Section 4.7). We conducted an additional in-depth evaluation in dataset D_2, assessing the segmentation performance for different tumor scales: small, medium, and large.

Number of central slices	267 (images), 267 (masks)
Size	210 MB
File Type	tif (image), tif (mask)
Resolution (in pixels)	512x512

Table 4.1: Data Specifications for D_1.

4.2 Dataset Description

In this work, we used two datasets: lung segmentation D_1 and lung cancer segmentation D_2. The dataset selection was inspired by Primakov et al. [PIVT+22a] workflow, where they first isolate lung area and then perform lung cancer segmentation. The datasets are a collection of both 2D and 3D CT scans, however, we have only utilized the 2D CT scans due to SAM's limitations that are explained in Section 4.5.

These two datasets are different in terms of their complexity. While dataset D_1 has a large target (lung), dataset D_2 is highly imbalanced with its small region of interest, i.e. lung cancer. This contrast between different targets gives us a broader sense of model performance.

4.2.1 Lung Segmentation Dataset D_1

The lung segmentation dataset D_1 includes manually segmented lung CT scans in both 2D and 3D formats. As previously mentioned, SAM is only compatible with 2D images. Therefore, we have focused on 2D CT central slices for this work. The dataset was obtained from Kaggle [Mad17]. We have $D_1 = \{p_1, \ldots, p_j\}$, $j \in \{1, \ldots, 267\}$, where j denotes running index of patients (image pairs). Image pairs are defined as $p_j = \{I_j, M_j\}$, where I_j and M_j represent j-th image and corresponding binary mask, respectively. Each binary mask consists of 0s for the background and 1s for the target. We have included dataset specifications in Table 4.1.

The metadata contains features such as lung area pixels, size in mm2, and percentile density. However, this was not used in this work because we only focused on the image segmentation task. Additional information such as CT vendors, reconstruction kernels, patient's health, annotators, or similar was not available. A single image pair example of D_1 dataset can be seen in Figure 4.2.

4.2.2 Lung Cancer Segmentation Dataset D_2

The lung cancer segmentation dataset D_2 consists of 96 non-small cell lung cancer patients with CT volumes from Stanford University. Originally, it was publicly accessible through TCIA [CVS+13]. However, we have used a preprocessed version from the Medical Segmentation Decathlon (MSD) challenge [ARB+22] wherein all images were reformatted from Digital Imaging and Communications in Medicine (DICOM) format to Neuroimaging Informatics Technology Initiative (NIfTI) images. Unlike dataset D_1, only

Number of slices	63 (images), 63 (masks)
Number of slices	2 x 17355
Number of positive slices	2 x 1597
Number of negative slices	2 x 15758
Size	6.09 GB
File Type	nii.gz (image), nii.gz (mask)
Resolution (in pixels)	512x512

Table 4.2: Data Specifications for D_2.

the training folder has ground truth masks available. Thus, we have used 63 NSCLC patients' CT volumes and split them into training, validation, and test sets. D_2 dataset CT volumes were performed pre-surgery with a section thickness of less than 1.5mm, a voltage of 120kVp, an adjustable tube current modulation ranging from 100 to 700mA, a tube rotation speed of 0.5 seconds, a helical pitch from 0.9 to 1.0, alongside a sharp kernel for image reconstruction [SAB+19]. The tumor area was manually segmented by an expert thoracic radiologist using OsiriX software. The basic specifications are listed in Table 4.2.

This dataset can be denoted as D_2 consisting of patient samples $p_j = \{V_j, VM_j\}$, with $j \in \{1, \ldots, 63\}$ representing the running index of patients, $V_j \in \mathbb{R}^3$ the image volume and $VM_j \in \mathbb{R}^3$ the corresponding binary mask. Each volume consists of i slices, denoted as $s_i = \{I_j^i, M_j^i\}$, where I_j^i represents the image slice and M_j^i corresponding binary mask for the j^{th} patient and i^{th} slice within that volume.

The lung cancer segmentation dataset was particularly interesting due to the complexity of segmenting a relatively small target (cancer) on a large image frame. Figure 4.3 shows a single pair example for D_2.

4.3 Data Preprocessing

In the initial lung segmentation dataset D_1, no preliminary data preprocessing was required since it only comprised the central slice for each patient's CT scan.

To ensure comparable outcomes to dataset D_1, we have used only the positive slices (containing tumor) from dataset D_2. For each patient p_j, we only consider 2D slice pairs containing a positive label: pairs where $\sum M_j^i > 0$, with i denoting the running index of 2D slices in the 3D volume. Figure 4.3 visualizes small-scale, medium-scale, and large-scale tumors from dataset D_2.

For CNN-based models, we used the original image size [512, 512] for both datasets. However, due to frequent crashes during training with this size on SAM, we adjusted the image size to [256, 256] for both datasets.

Regarding image scaling, I_j is resized using the bicubic interpolation algorithm, however, we use the resize nearest neighbor interpolation algorithm for M_j. The nearest neighbor

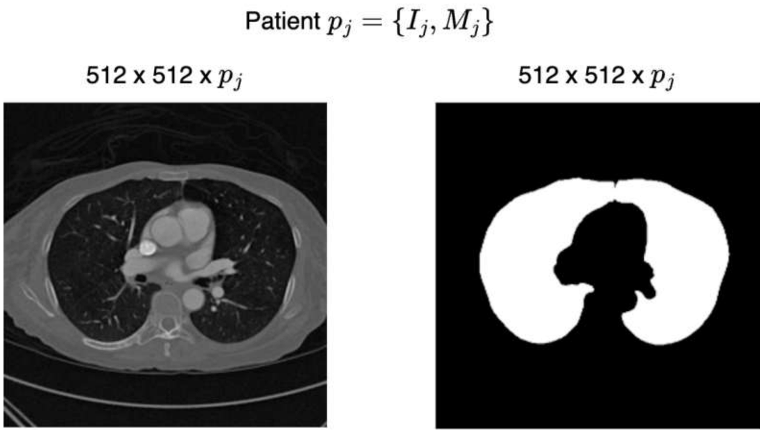


Figure 4.2: Dataset D_1 contains a central lung slice from CT scan for each I_j. Therefore, as each slice contains pixels annotated as lung $\sum M_j > 0$ for patient p_j

algorithm is a resampling method that matches the closest pixel value from an original image to a down-scale image.

4.3.1 Data Augmentation

Similarly, as Primakov et al. [PIVT+22a], we apply data augmentation techniques, on a training set only, to ensure the robustness of the model. It is an effective method to improve the generalization of the model by enriching training data [YXZ+23]. This process applies a series of transformations to an image such as flipping, rotating, and scaling.

We use a sequential order of transformations, which is listed as follows:

1. **Horizontal Flip** receives an input x_h=`0.5` that defines a likelihood to flip the image horizontally.

2. **Vertical Flip** similarly as horizontal flip, it receives parameter x_v=`0.5` to perform a vertical flipping.

3. **Linear Contrast** regulates contrast by scaling image pixels within a certain input range α, in our case uniformly sampled from `(0.75, 1.25)`.

$$L_{contrast} = 127 + \alpha(v - 127) \tag{4.1}$$

   where v denotes a pixel value [aug24].

4. **Affine** applies the following transformations: translation, scaling, and rotation. The translation shifts (translates) an image by a parameter t=`0.15` on the x-axis or y-axis, relative to their size. Scaling receives an input range, `(0.85, 1.15)` in

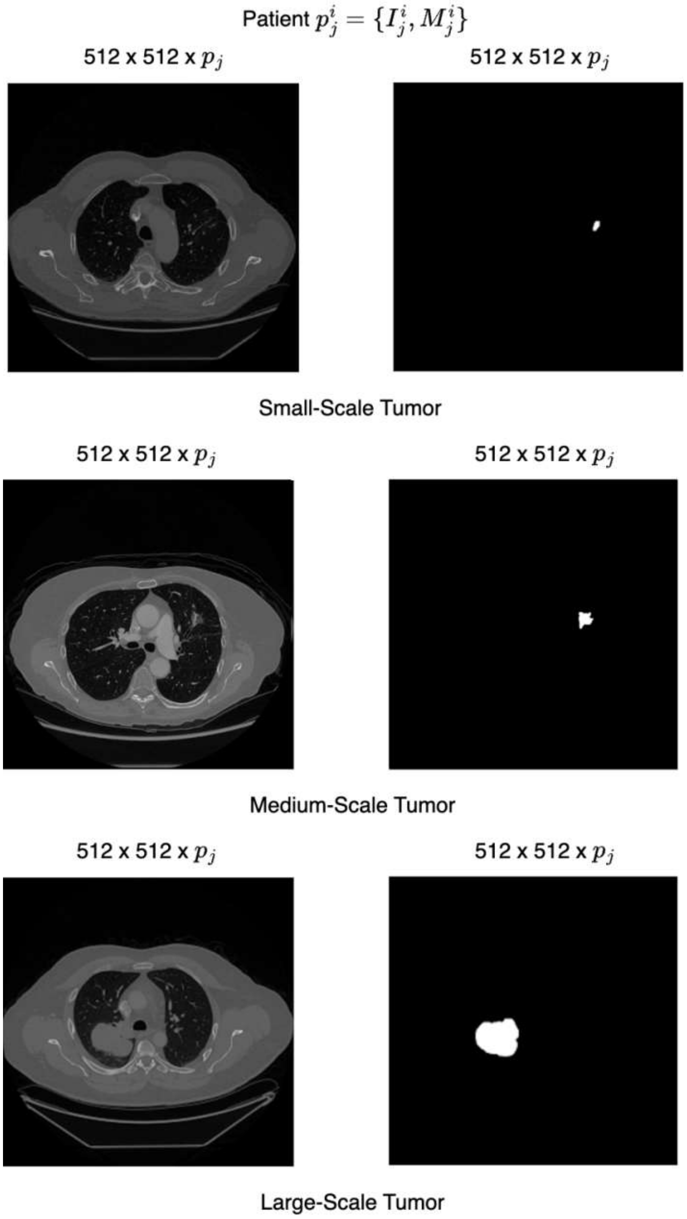


Figure 4.3: Dataset D_2 contains lung cancer CT scans. Starting from the top, the first pair shows a small-scale tumor, the second pair is a medium-scale tumor, and the third pair displays a large-scale lung cancer slice for patient p^i_j.

relative size, to either downscale or upscale an image. Rotation receives a parameter $[-x, x]$ from uniformly sampled range `(-45, 45)` that rotates an image.

5. **Elastic Transformation** is the last transformation in this sequence that applies

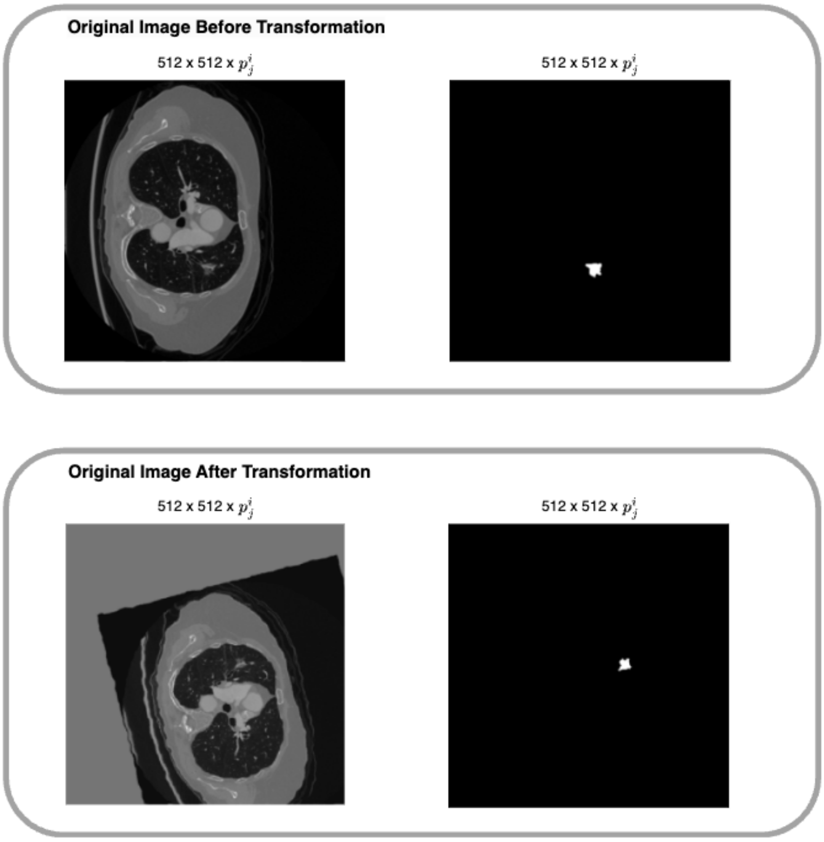


Figure 4.4: Lung cancer p_j^i CT scan before and after transformation used for data augmentation. The transformations include horizontal flip, vertical flip, linear contrast, and elastic transformation.

image deformations using a displacement field. It was used with default parameters α and σ which regulate displacement (Figure 4.4).

4.3.2 Data-Specific and Model-Specific Preprocessing

This part addresses both data-specific and model-specific preprocessing methods used in this work. The reason for data-specific preprocessing is lung segmentation dataset is an easy target compared to lung cancer segmentation which is very complex to segment due to its small region of interest. To avoid redundant descriptions, we focused only on the differences in preprocessing.

1. Lung Segmentation Dataset

There was no preliminary data-preprocessing for dataset D_1 since each $\sum M_j > 0$ for patient p_j. In other words, it only includes positive slices that contain the target.

A. CNN-based models
We split the dataset into 60% training, 10% validation, and 30% testing sets.

B. SAM-based models
For each image pair $p_j = \{I_j, M_j\}$ we construct a bounding box using the ground truth as explained in Section 4.5. We replicate the same process for the Segment Anything Model without human interaction, however, the bounding box is set to match the image size. This allows us to avoid using ground truth for the Segment Anything Model.

2. Lung Cancer Segmentation Dataset

We have focused on lung cancer-positive slices to compare the CNN model to SAM. As already noted, SAM is limited to 2D images and must contain the target of interest. For our experiments, we only extracted original size [512, 512] 2D slices with manual cancer annotations where $\sum M_j^i > 100$.

A. CNN-based models
We included image intensity scaling and data augmentation, however, it was not used in other settings. The intensity preprocessing function standardizes the intensity range of the pixel values from a given image. It takes five parameters: image, a_{min}, a_{max}, b_{min} and b_{max}. In the first step, we normalize image intensity values by using a_{min}=`-57` and a_{max}=`164` range from MONAI spleen segmentation example [CLB+22][spl24].

$$I_{jnorm}^i = \frac{I_j^i - a_{min}}{a_{max} - a_{min}} \tag{4.2}$$

Subsequently, we rescale the normalized image to the desired range from b_{min}=`0` and b_{max}=`1`.

$$I_{jnew\_range}^i = I_{jnorm}^i * (b_{max} - b_{min}) + b_{min} \tag{4.3}$$

Lastly, we implement the clip method to limit the pixel values to the range defined by b_{min} and b_{max}, similar as intensity function from MONAI spleen segmentation example [CLB+22][spl24].

The dataset is split patient-wise and randomly into 80% training, 10% validation, and 10% testing set, based on the assumption that the training would require more data than dataset D_2 because of its small target size.

B. SAM-based models
We use identical data split from CNN-based models. Everything else is the same as D_1 dataset SAM preprocessing.

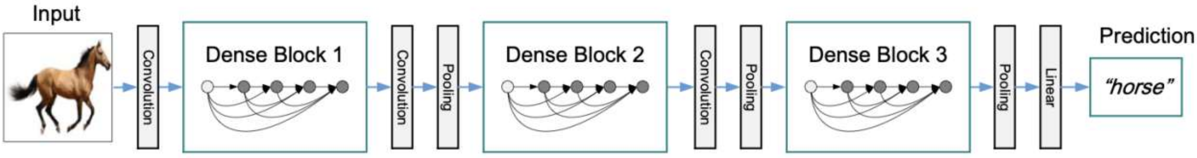


Figure 4.5: Triple Block DenseNet with Transition Layers. It receives an input image and uses convolution for feature extraction. After that, there are three dense blocks with transition layers that consist of convolution and pooling. Finally, the linear transition layer outputs a prediction [HLvdMW18].

4.4 CNN-based Models

In this section, we present the CNN-based models applied to both datasets D_1 and D_2. This work involves experimenting with different model architectures, backbones (network encoders), and loss functions. More focus was given to dataset D_2 due to its increased complexity compared to dataset D_1. A full list that specifies each configuration used for CNN-based models is given in Table 4.3.

4.4.1 Encoder Networks

In this work, we have used some of the most prominent backbones including ResNet, EfficientNet, and DenseNet.

Introduced by He et al. [HZRS15], Residual Networks (ResNets) are known for their *skip connections* or *residual connections*. These connections allow for the training of deeper networks than previously feasible. ResNets are available in various scales and depths, such as ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152, where the number represents the total number of layers. In this work, we used ResNet18, ResNet34, ResNet101, and ResNet152.

Tan et al. [TL19] introduced EfficientNets that uniformly scale all dimensions of the network (depth, width, and resolution). The foundational EfficientNet-B0 serves as a scalable building block for the B1-B7 EfficientNet variants. The model complexity increases as variants progress from B1 to B7 EfficientNet. EfficientNet-B7 set a new record in 2020 for accuracy on ImageNet. We only used EfficientNet-B7 in our experiments.

In a Densely Connected Convolutional Network (DenseNet), each layer is directly connected to every other layer in a feed-forward fashion (Figure 4.5). As described by Huang et al. [HLvdMW18], such a dense connectivity pattern reduces the need for capturing redundant feature maps, therefore, it improves efficiency. In this work, we've only used the DenseNet121 encoder, which has the lowest number of parameters compared to other DenseNet encoders such as DenseNet161 or DenseNet169.

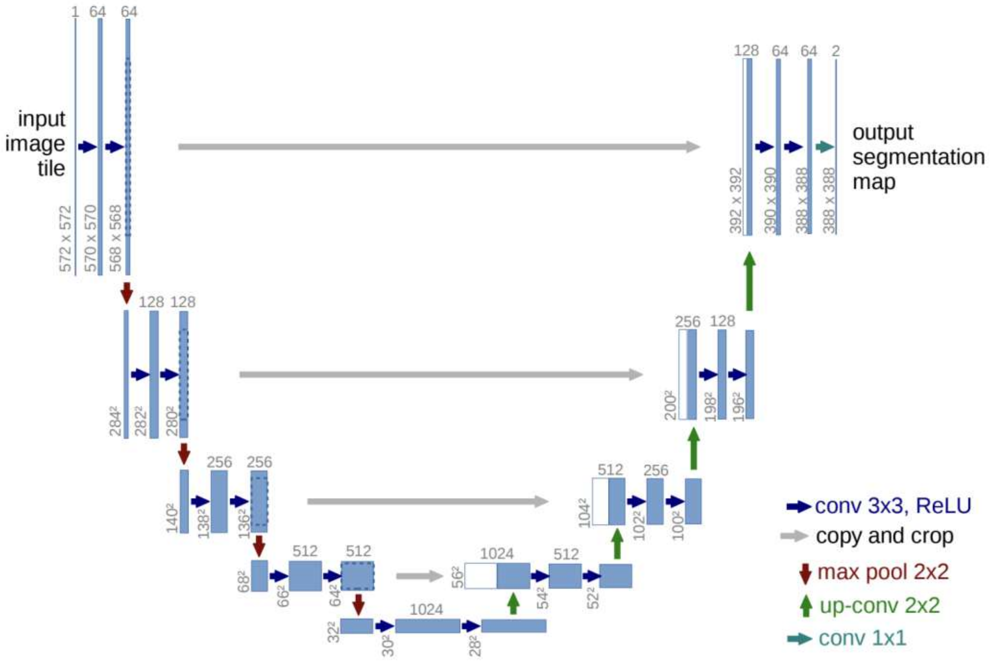


Figure 4.6: U-Net Model Architecture. It consists of four encoders (left side) and four decoders (right side). The last convolution layer uses an activation function to generate a prediction for each pixel [RFB15].

4.4.2 Network Architectures

When developing the model for semantic segmentation, it is crucial to experiment with different model architectures and backbones since the results vary across various tasks and datasets [Gup23].

In this work, we used four different network architectures for image segmentation including original U-Net, U-Net++, Feature Pyramid Network (FPN), and Pyramid Scene Parsing Network (PSPNet). These network architectures can be configured with various encoder networks or backbones.

U-Net originally consisted of four encoder and four decoder blocks connected in a U-shape as illustrated in Figure 4.6. Finally, 1x1 convolution with activation function is used to generate pixel-wise prediction [RFB15].

Furthermore, we applied U-Net++ on the dataset D_2. The U-Net++ is an enhanced version of the U-Net architecture. It is distinguished by re-designed nested and dense skip pathways which minimize the semantic differences between the feature maps of the encoder and decoder (Figure 4.7). We decided to use U-Net++ in addition to the U-Net because it achieves better performance on various medical image segmentation tasks such as nodule isolation in chest low-dose CT scans, nuclear segmentation in microscopic

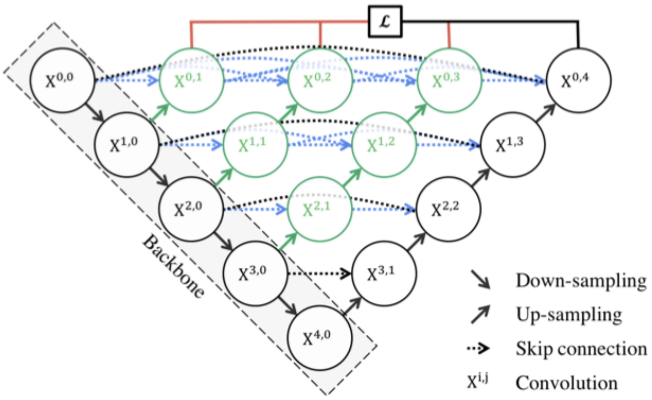


Figure 4.7: U-Net++ Model Architecture. Its encoder and decoder are connected with a series of nested dense convolutional blocks to minimize the semantic gap [ZSTL18].

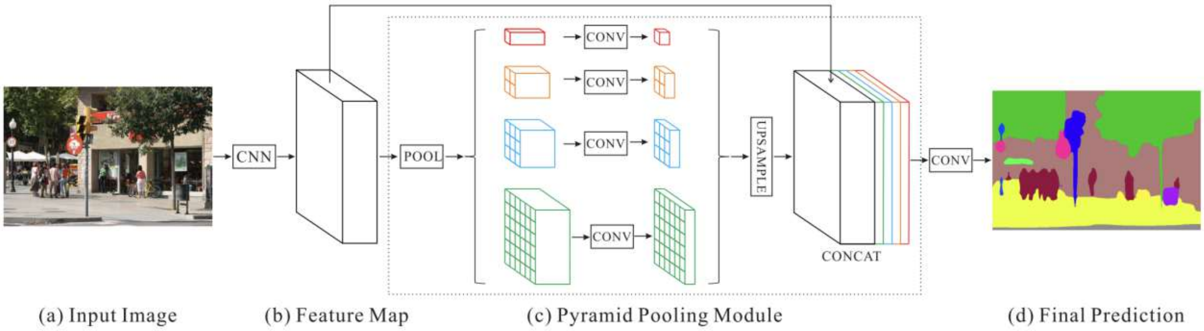


Figure 4.8: Pyramid Scene Parsing Network (PSPNet) Model Architecture. (a) The input image is (b) encoded into a feature map, which (c) is further processed with four pyramid scales that extract different sub-regions to generate (d) the final prediction [ZSQ+17].

images, or liver segmentation in abdominal CT [ZSTL18].

The PSPNet architecture creates a feature map from an input image using CNN and transfer learning, after that it uses a pyramid pooling module (Figure 4.8). The pyramid pooling module captures features using four different pyramid scales, and outputs concatenated base feature maps as the final global feature [ZSQ+17].

The FPN model stands out with its top-down architecture and lateral connections that allow for high-level semantic segmentation (Figure 4.9). Its architecture allows for combining low resolution for semantically weak features, and high resolution for semantically important features [LDG+16].

Lastly, we have used original U-Net, U-Net++, PSPNet, and FPN in this work [Iak19].

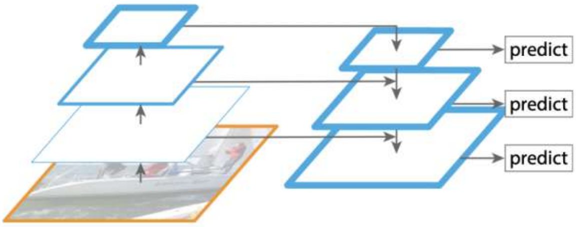


Figure 4.9: Feature Pyramid Network (FPN) Model Architecture. It takes an input image to build a feature pyramid for each scale. Its top-down pathway and lateral connections allow for capturing important semantic features with high resolution [LDG+16].

A hyper-parameter search space overview for our models is available in Table 4.3.

4.4.3 Loss Functions

In this work, we have used *Binary cross-entropy* and *Dice Similarity Coefficient* loss functions.

Our Binary cross-entropy loss was wrapped with a sigmoid function in the last layer to ensure that predictions are bound between 0 and 1; background or target. The binary cross-entropy loss measures the difference between predictions and ground truth segmentation using a pixel-wise comparison. It is given by:

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \tag{4.4}$$

where n represents the number of pixels in an image. y and $\hat{y}$ represent ground truth and predicted segmentation map for each pixel, respectively [TCERPCU23].

Based on the Dice Similarity Coefficient (DSC), explained in Subsection 2.5.1, the Dice loss maximizes the similarity between the predictions and ground truth segmentation mask. This loss function is given by:

$$L_{DL} = 1 - \frac{2|y \cap \hat{y}|}{|y| + |\hat{y}| + \epsilon} \tag{4.5}$$

where $\epsilon = 1 \times 10^{-7}$ was used for numerical stability. Inspired by Primakov et al. [PIVT+22a], we combined widely used BCE and Dice Loss. Each of them was equally weighted in a single loss function. It can be denoted as follows:

$$L_{BCE\_DL} = w1 \cdot L_{BCE} + w2 \cdot L_{DL} \tag{4.6}$$

where w_i represents weights. In our case, this was set to 0.5 for both weight w_1 and weight w_2.

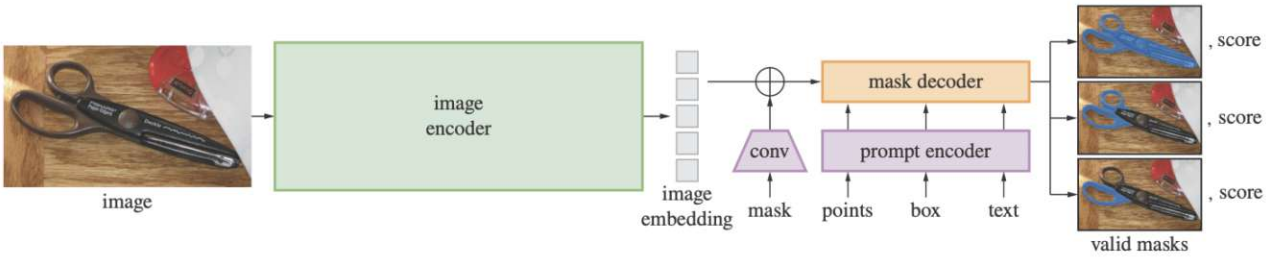


Figure 4.10: Segment Anything Model Key Components. It uses an image encoder to create image embedding which can be queried by input prompts such as points, box, or text prompt. In case of an ambiguous prompt, SAM can output more than one mask [KMR+23].

4.5 SAM-based Models

In the comparative analysis with CNN-based models, we have used the Segment Anything Model (SAM) that is based on Vision Transformer (ViT) explained in Section 2.4. It implies being used with prompts including text, points, masks, or bounding boxes. In our experiments, the ground truth was utilized for SAM with SHI, however, it poses a challenge for a direct comparison with CNN-based models. Nevertheless, SAM was fine-tuned without SHI to ensure a fair comparison ([Rog23],[Bha23]). Figure 4.12 shows the bounding box prompt comparison between SAM with and without SHI.

SAM consists of three key components: image encoder, prompt encoder, or mask encoder (Figure 4.10). An image encoder uses a pre-trained ViT masked autoencoder [HCX+21] as a backbone to embed the input image. The prompt encoder may receive four types of prompts: text, points, masks, and bounding boxes. A lightweight mask decoder uses self-attention (single input sequence) and cross-attention (combines multiple input sequences) to combine an image and prompt embedding. After that, it utilizes a dynamic linear classifier, which automatically adjusts classifier criteria, to output a prediction, or probability map [KMR+23]. SAM only supports two-dimensional images as an input.

Moreover, there are three SAM backbones available: SAM-ViT-Base, SAM-ViT-Large, and SAM-ViT-Huge with 91M/308M/636M parameters, respectively. Obviously, the SAM-ViT-Base is the smallest network and overall performs very close to the SAM-ViT-Huge according to Kirillov et al. [KMR+23]. Thus, their differences will not be analyzed in contrast to the CNN's encoder networks from the section above. We have utilized a pre-trained SAM model and fine-tuned its lightweight encoder. Furthermore, we have utilized a loss function that integrates Dice Loss and Cross Entropy Loss which was already explained in Subsection 4.4.3. Adam was chosen as the optimizer for this task.

4.5.1 SAM with Simulated Human Interaction

SAM was fine-tuned with simulating human interaction through a bounding box prompt (Figure 4.11). We have chosen the bounding box type of prompt since it clearly selects ROI and reduces ambiguity compared to point prompt which often requires multiple interactions [MHL+23]. In our preliminary analysis, we tried using text prompts, however, it did not lead to reasonable results. For simulated human interaction, we have calculated the bounding box (BB) coordinates for a given ground truth segmentation map. Here is a step-by-step explanation.

For each data point:

1. Based on the ground truth segmentation map compute the coordinates of the smallest rectangle that encompasses the target structures (defined by non-zero pixels).
2. To simulate human interaction, add some randomness to the bounding box coordinates. For each coordinate `(x_min, x_max, y_min, y_max)` sample a random integer value between `0` and `x` from discrete uniform distribution. We utilized `x=5` and `x=20` to produce random `(0, x)`. These ranges were defined during our initial analysis since they simulate a more realistic human interaction scenario. A random integer `x` needs to be positive to prevent target cutting.
3. Finally, create the bounding box of the segmented object with some added randomness $rand_i$.

$$(x\_min - rand_1, x\_max + rand_2, y\_min - rand_3, y\_max + rand_4) \tag{4.7}$$

This process of prompting or simulating human interaction is applied for both fine-tuning the SAM model and segmenting new unseen instances.

4.5.2 SAM without Simulated Human Interaction

SAM without SHI is not the typical application of this model. However, to ensure a fair comparison with the CNN-based models (Section 4.4), any information leak from the ground truth must be excluded from the fine-tuning and evaluation step. The implementation is almost identical to the SAM with SHI, the only difference is that we used bounding boxes covering the full image as prompt input. For example, if the image size is 256x256 the BB would be given as follows:

$$(x\_min, x\_max, y\_min, y\_max) \tag{4.8}$$

$$(0, 256, 0, 256). \tag{4.9}$$

Through this method, we can fine-tune the model utilizing all image pixels, unlike SAM with the SHI approach, where only a small region of the entire image is analyzed.

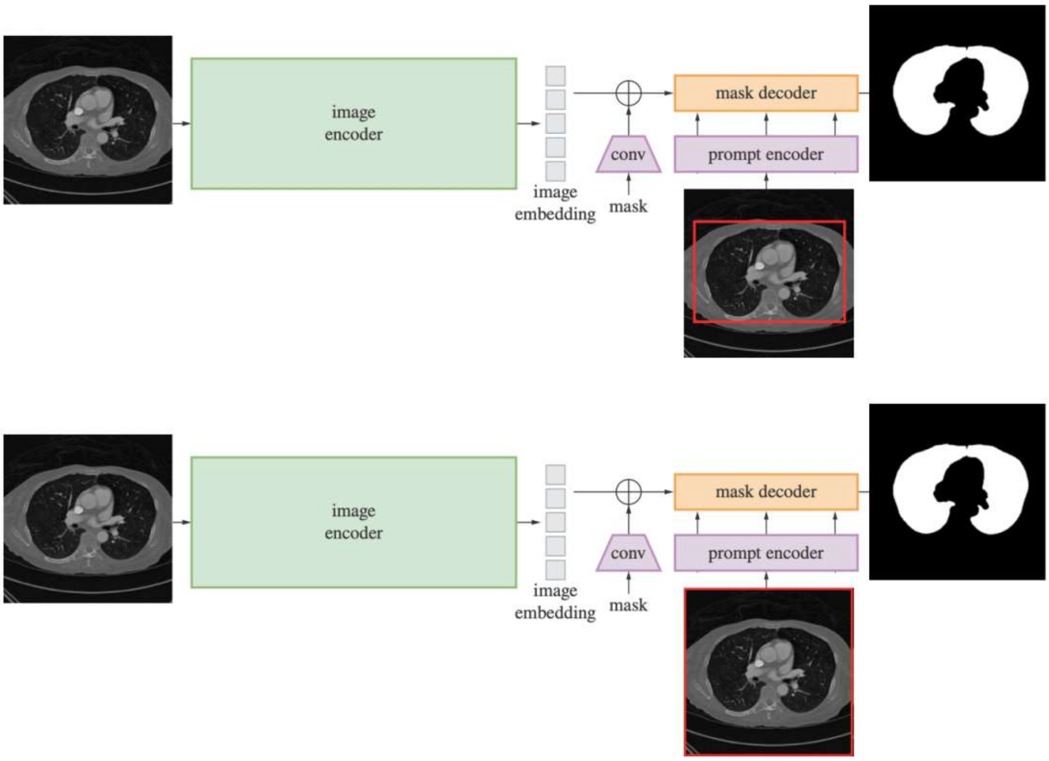


Figure 4.11: Key components of Segment Anything Model. The top part, visualizes SAM with simulated human interaction, while the bottom part visualizes SAM without simulated human interaction. [KMR+23]

4.6 Hyperparameter Tuning and Model Selection

In this work, we experimented with various models, architectures, and backbones. During the initial phase, we tried using different loss functions, optimizers, and number of epochs. Table 4.3 describes the search space for our experiments.

For the lung segmentation dataset D_1, we utilized an SGD optimizer at a 0.001 learning rate and Binary Cross Entropy loss function across all CNN-based models. We used U-Net model architecture combined with resnet18/34/152 and efficientnet-b7. Also, we conducted training with 50 and 200 epochs, and the best model was chosen based on the validation set DSC and HD performance. SAM, both with SHI and without SHI were trained with 50 epochs and three different backbones including ViT-Base, ViT-Large, and ViT-Huge. An Adam optimizer set to a learning rate of 1×10^{-5} and the Dice combined with the Cross Entropy Loss function was employed. SAM with SHI was trained and evaluated using `x=5` and `x=20` bounding boxes.

We applied U-Net/U-Net++/FPN/PSPNet model architectures with resnet18/101/152, efficientnet-b7, and densenet121 network encoders on lung cancer segmentation dataset

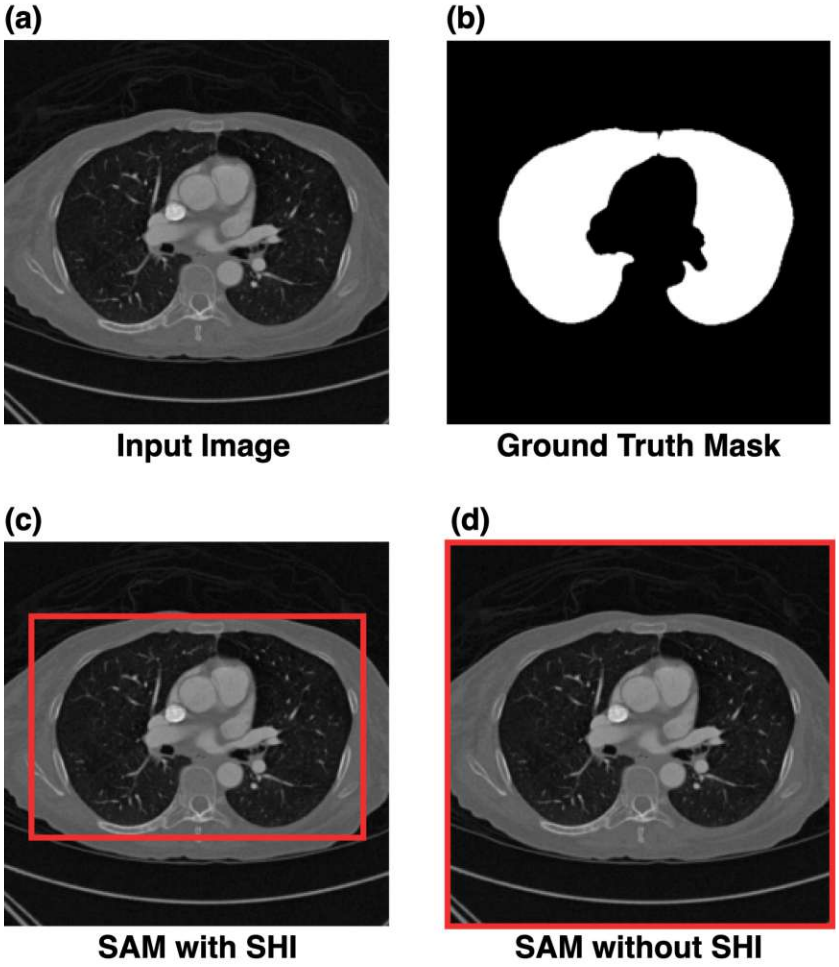


Figure 4.12: The input image **(a)** and ground truth mask **(b)** represent an image pair $p_j = I_j, M_j$ from dataset D_1. SAM with simulated human interaction has a corresponding bounding box provided in the bottom left corner **(c)**, while a full-size bounding box from SAM without simulated human interaction is in the bottom right corner **(d)**.

D_2. The AdamW optimizer with a learning rate set to 0.001 and a loss function that combines BCE and Dice Loss was used across all CNN-based models. We trained CNN-based models with 100 and 200 epochs. The configuration for SAM with SHI and without SHI remained consistent with the approach from dataset D_1.

These parameters were determined through a series of preliminary experiments and demonstrated optimal performance based on the validation set.

Parameter	**Search Space**
CNN Model	
Model Architectures	U-Net, **U-Net++, **FPN, **PSPNet
Network Encoders	resnet18/*34/**101/152, efficientnet-b7, **densenet121
Optimizer	*SGD(lr=0.001), **AdamW(lr=0.001)
Loss Function	*Binary Cross Entropy (BCE), **Combined BCE and Dice Loss
Number of epochs	*50, **100, 200
SAM with SHI	
Backbones	SAM-ViT-Base, SAM-ViT-Large, SAM-ViT-Huge
Bounding Box Size (in pixels)	5, 20
Number of epochs	50
SAM without SHI	
Backbones	SAM-ViT-Base, SAM-ViT-Large, SAM-ViT-Huge
Bounding Box (BB)	img_dim
Number of epochs	50

Table 4.3: Hyperparameter tuning and model selection search space table. Non-starred parameters were used for both datasets. A single asterisk parameters were used for dataset D_1, and those with double asterisks for dataset D_2. Img_dim represents the full image dimension.

4.7 Experimental Setup

We used DSC, HD, standard deviation (SD), and 95th percentile (H95th) as quantitative evaluation metrics in our experiments. Furthermore, we evaluate dataset D_2 for each tumor scale: small, medium, and large. These scales are determined based on training data distribution.

Dataset D_1 and dataset D_2 are evaluated on their test sets using DSC and HD metrics. Additionally, we compute standard deviation (SD), and 95th percentile (H95th), for both DSC and HD. Both CNN-based and SAM-based models output a probability map as a prediction, which is converted to a binary segmentation map using a threshold of 0.4 and 0.5 for CNN-based models and SAM-based models, respectively. These thresholds were determined for each model architecture in preliminary experiments on the validation set.

Dataset D_2 positive CT scans were divided into three uniform subgroups based on an analysis of the training data distribution. The tumor scales are defined as follows:

1. **Small Scale Tumor**: $[\sum M_j^i > 100, \sum M_j^i < 500]$
2. **Medium Scale Tumor**: $[\sum M_j^i > 500, \sum M_j^i < 1500]$
3. **Large Scale Tumor**: $\sum M_j^i > 1500$.

For each tumor scale test set, we evaluated the DSC and HD for CNN-based and SAM-based models. This approach of evaluating tumors across different scales is widely adopted in similar research papers such as Primakov et al. [PIVT+22a].

4.8 Training Details

In this work, we conducted our experiments at the HPC cluster of the CIR lab, running a Linux CentOS operating system with 9 servers. Most of the experiments were done using a specific server on the HPC cluster with 20 CPU cores, and 6 x GeForce RTX 2080 Ti GPUs with 11GB of RAM with CUDA v11.7. However, we have also used NVIDIA GeForce RTX 3080 Ti and Nvidia TITAN Xp with 12 GB of RAM from other nodes. We used Slurm Workload Manager to automatically schedule the experiments efficiently on the servers.

Python (v3.6.8) was used for CNN models and Python (v3.11.0) for SAM. PyTorch (v1.10.1) was used as a deep learning framework that supports GPU acceleration [PGM+19]. For CNN-based models implementation, we utilized Segmentation Models PyTorch library (v0.3.1) with pre-trained backbones on ImageNet [Iak19].

For tracking the experimental results, we have used Weights&Biases (v0.15.11). We used the 3D Slicer (v5.2.2) open-source app for MacOS to preview CT scans in NIFTI format.

CHAPTER 5

Results

This chapter presents results for lung segmentation in Section 5.1, lung cancer segmentation in Section 5.2, and lung cancer segmentation for small, medium, and large scale in Section 5.3 for CNN-based models, SAM with SHI and SAM without SHI. Finally, this section includes both quantitative and qualitative metrics for both dataset D_1 and D_2 test sets.

5.1 Lung Segmentation Results

In this section, we present results that show a comparison between the performance of CNN-based and SAM-based models on the lung segmentation dataset D_1 test set. We report mean, H95th, and SD for DSC and HD metrics.

5.1.1 CNN-based Models

An overview of quantitative results across all CNN-based models applied to lung segmentation dataset D_1 with mean DSC and mean HD on the test set is provided in Table 5.1. We can see consistent performance in both DSC and HD for test sets across different models. The highest DSC was 0.975 using U-Net with efficientnet-b7, and the same model achieved the lowest HD of 5.859. Moreover, these results demonstrate that there was no need to explore additional model architectures, given the satisfactory performance with current configurations.

The training time duration had significant variation from approximately 4 minutes for U-Net with a ResNet 18 to around 1 hour for U-Net with ResNet 152. In addition, we report the number of epochs used for CNN-based model training.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	efficientnet-b7	50	**0.975**	**0.026**	**0.054**	**5.859**	**3.694**	**1.539**
U-Net	resnet18	50	0.975	0.028	0.055	5.907	3.681	1.560
U-Net	resnet34	50	0.973	0.030	0.055	6.084	3.503	1.481
U-Net	resnet152	50	0.970	0.029	0.063	6.017	3.826	1.739
U-Net	resnet152	200	0.974	0.026	0.055	6.020	7.874	3.515

Table 5.1: CNN Model Performance on Lung Segmentation Dataset D_1 Test Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	0.964	0.057	0.021	4.959	3.253	0.991
sam-vit-large	5	0.961	0.078	0.037	4.936	2.650	0.965
sam-vit-huge	5	0.961	0.065	0.038	4.899	2.481	0.977
sam-vit-base	20	**0.966**	**0.052**	**0.018**	**4.805**	**2.815**	**0.818**
sam-vit-large	20	0.965	0.060	0.019	4.913	2.622	0.862
sam-vit-huge	20	0.957	0.072	0.047	4.954	2.868	0.976

Table 5.2: SAM with SHI Performance on Lung Segmentation Dataset D_1 Test Set.

5.1.2 SAM with Simulated Human Interaction

The results presented in Table 5.2 for the lung segmentation dataset D_1 showcase the performance of different SAM backbones with varying bounding box sizes. These results cannot be directly compared to CNNs since we are already using the ground truth for the bounding box to simulate human interaction. Therefore, the model's performance is partly dependent on initial human input.

The results in Table 5.2 for dataset D_1 using SAM show relatively high performance across all model's backbones with DSC being 0.957 or greater. The performance across different backbones remains similar. Also, the model results are consistent for both `x=5` and `x=20` bounding box sizes.

Moreover, the HD is generally reduced relatively to the results using CNNs in Table 5.1 with an absolute mean difference of 1.06. This suggests that the lungs are captured better using SAM.

5.1.3 SAM without Simulated Human Interaction

It is important to note that SAM is not intended to be used without human interaction or a full-sized bounding box which has been explained in Section 4.5. The results presented in Table 5.3 for the lung segmentation dataset D_1 using SAM without human interaction can be directly compared to the CNN-based models approach on D_1. The DSC varies significantly across different SAM backbones compared to CNN-based models. Based on

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.831	0.285	0.120	7.577	3.379	1.108
sam-vit-large	0.897	0.134	0.091	6.967	3.189	1.153
sam-vit-huge	**0.920**	**0.137**	**0.075**	**5.749**	**3.463**	**1.254**

Table 5.3: SAM without SHI Performance on Lung Segmentation Dataset D_1 Test Set.

DSC and HD, SAM with ViT-Huge backbone performs similarly to CNN-based models in Table 5.1 with a DSC absolute mean difference of 0.091. Lastly, we provide boxplot and qualitative results that include the best CNN-based and SAM-based models in Figure 5.1 and Figure 5.2, respectively.

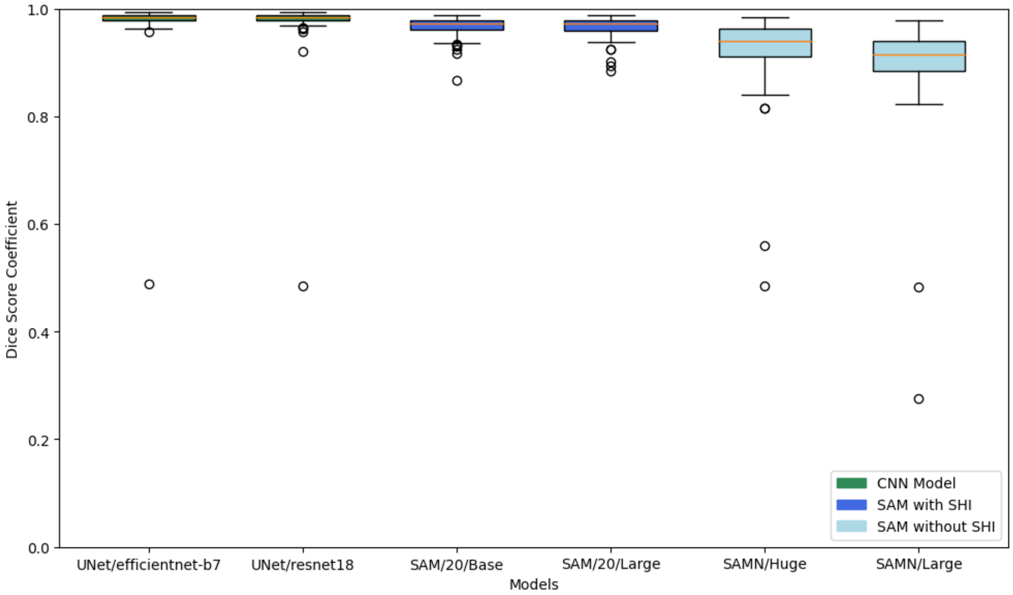


Figure 5.1: Boxplot of Two Best Performing Models on Lung Segmentation Dataset D_1 Test Set Based on DSC. The orange line indicates the median, and the whiskers extend DSC values from minimum to maximum.

5.2 Lung Cancer Segmentation Results

In this section, we report the results for the lung cancer dataset D_2 test set using CNN-based models, SAM with SHI, and SAM without SHI. As previously stated, this dataset is far more complex compared to lung segmentation due to its small target, which is a common challenge in the field of medical image analysis.

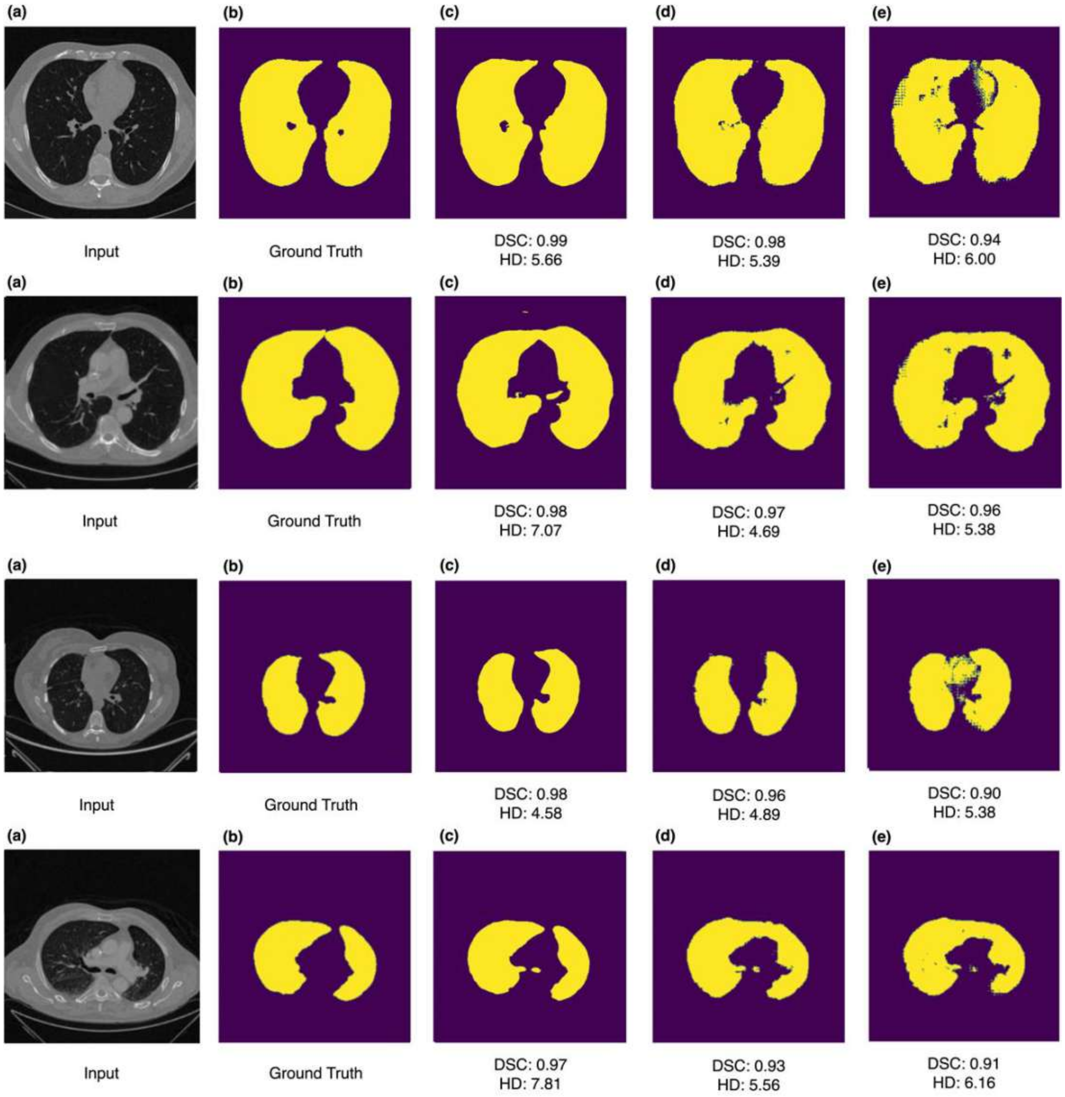


Figure 5.2: Qualitative Results of Lung Segmentation D_1 Dataset Test Set. The input image **(a)** and ground truth mask **(b)** represent an image pair $p_j = I_j, M_j$ from dataset D_1. Images **(c)**, **(d)**, and **(e)** represent the output from the best performing CNN-based model (UNet/efficientnet-b7), SAM with SHI (SAM/20/Base) and SAM without SHI (SAMN/Huge), respectively.

5.2.1 CNN-based Models

CNN-based model results for lung cancer segmentation are shown in Table 5.4. The highest DSC of 0.440 was achieved using U-Net with resnet101. SD is very high across all models, however, HD results are relatively similar.

The U-Net++ with a single dagger (†) uses data augmentation to improve the model

			DSC			**HD**		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.428	0.896	0.334	4.651	5.832	1.882
U-Net	efficientnet-b7	100	0.387	0.893	0.356	4.740	5.250	1.839
U-Net	resnet18	100	0.372	0.888	0.346	4.707	5.680	1.712
U-Net++	efficientnet-b7	100	0.390	0.869	0.347	4.339	4.483	1.489
FPN	resnet18	100	0.318	0.850	0.336	**4.297**	**4.262**	**1.340**
PSPNet	efficientnet-b7	100	0.289	0.777	0.292	4.613	4.001	1.362
U-Net++	resnet152	100	0.400	0.870	0.351	4.973	7.515	2.296
†U-Net++	resnet152	100	0.234	0.758	0.279	6.865	9.491	3.040
U-Net	densenet121	100	0.331	0.818	0.305	5.565	9.095	2.371
††U-Net	resnet18	100	0.178	0.707	0.260	4.544	3.980	1.221
U-Net++	efficientnet-b7	200	0.392	0.849	0.319	4.489	4.557	1.476
U-Net	resnet101	100	**0.440**	**0.901**	**0.338**	4.605	5.312	1.736

Table 5.4: CNN Model Performance on Lung Cancer Segmentation Dataset D_2 Test Set.

results. Also, double-daggered (††) U-Net uses preprocessing with intensity scaling which is explained in Section 4.3. However, both techniques did not improve the overall score. In contrast, the model which used intensity scaling had the lowest DSC on the test set. Moreover, incorporating data augmentation into the model did not achieve increased robustness, as the standard deviation (SD) is also high.

In Table 5.4, results are not consistent across different configurations. For example, PSPNet has a much worse DSC of 0.289 compared to U-Net using resnet101 with a DSC of 0.440. Also, we have achieved worse overall performance compared to the previous task with lung segmentation dataset D_1 in Table 5.1. The FPN model achieved the lowest HD of 4.297 on the test set indicating the closest alignment between the predicted and the actual segmentation.

The training of the U-Net++ with ResNet152, alongside data augmentation, required about 24 hours of training, which is one of the highest in runtime duration.

5.2.2 SAM with Simulated Human Interaction

Results for lung cancer segmentation using SAM with Simulated Human Interaction are presented in Table 5.5. The results in Table 5.5 are better than in Table 5.4 with a DSC absolute mean difference of 0.322. SAM with ViT-Base backbone with x=5 bounding box achieves the highest DSC of 0.749, and the lowest HD of 2.084 on the test set. Also, x=5 bounding box size improves the model's robustness compared to x=20 bounding box size which is expected because the target is larger.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.749**	**0.552**	**0.173**	**2.084**	**1.748**	**0.560**
sam-vit-large	5	0.733	0.640	0.192	2.097	1.585	0.524
sam-vit-huge	5	0.746	0.656	0.199	2.098	1.810	0.596
sam-vit-base	20	0.656	0.842	0.247	2.266	1.924	0.632
sam-vit-large	20	0.619	0.809	0.253	2.327	1.924	0.651
sam-vit-huge	20	0.612	0.867	0.276	2.389	2.327	0.726

Table 5.5: SAM with SHI Performance on Lung Cancer Segmentation Dataset D_2 Test Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.053	0.355	0.144	3.427	2.877	0.864
sam-vit-large	0.061	0.3960	0.143	3.526	2.763	0.884
sam-vit-huge	**0.113**	**0.593**	**0.202**	**3.267**	**2.582**	**0.745**

Table 5.6: SAM without SHI Performance on Lung Cancer Segmentation Dataset D_2 Test Set.

5.2.3 SAM without Simulated Human Interaction

In this experiment, we present results for the lung cancer segmentation dataset D_2 test set in Table 5.6. The best-performing model was SAM with ViT-Huge backbone which achieved a very low DSC of only 0.133 on the test set, SAM with ViT-base backbone and SAM with ViT-Large backbone performed even worse. Also, SAM with ViT-Huge backbone scored the lowest HD of 3.267 on the test set.

These scores are lower compared to CNN-based with a DSC of 0.440 using U-Net from Table 5.4 and SAM with SHI with a DSC of 0.749 from Table 5.5.

Finally, we provide boxplot and qualitative results for CNN-based and SAM-based models in Figure 5.3 and Figure 5.4, respectively.

5.3 Tumor-Scale Based Lung Cancer Segmentation Results

In this section, we provide quantitative results for each tumor scale to assess the CNN-based, SAM with SHI, and SAM without SHI model performance across three different tumor scales including small, medium, and large scale test sets. The segmentation of small-scale tumors proved to be the most difficult task due to the complex target, SAM with SHI achieved the highest DSC of 0.704, significantly outperforming the best CNN-based model U-Net++ with efficientnet-b7 that achieved the DSC of only 0.327 (Subsection 5.3.1). The results based on medium-scale tumor improved across all models, the highest DSC of 0.804 was achieved using SAM with SHI (Subsection 5.3.2). Finally,

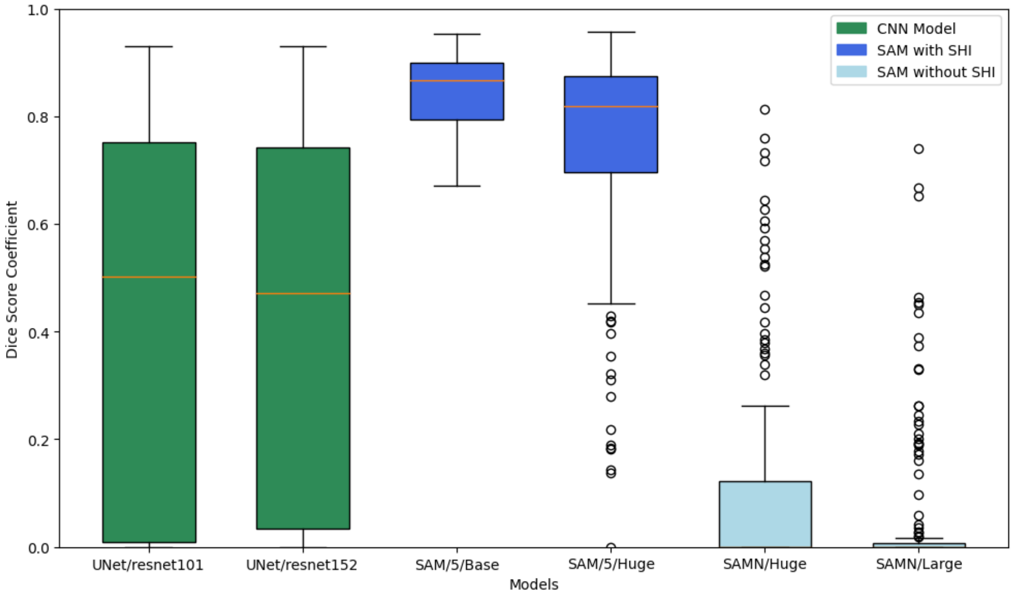


Figure 5.3: Boxplot of Two Best Performing Models on Lung Cancer Segmentation Dataset D_2 Test Set Based on DSC. The orange line indicates the median, and the whiskers extend DSC values from minimum to maximum.

large-scale tumor results improved for both CNN-based and SAM-based models, where the best CNN-based model (U-Net/resnet18), SAM with SHI, and SAM without SHI achieved the DSC of 0.771, 0.868, and 0.336, respectively (Subsection 5.3.3).

5.3.1 Small Scale Lung Cancer

The small-scale tumor was the most complex to segment since it is highly imbalanced.

CNN-based Models Each model from Table 5.7 has lower DSC on a small scale compared to general results on the test set. U-Net++ achieved the highest DSC of 0.327 on the test set, however, its performance dropped significantly with the usage of data augmentation to DSC of 0.132 or the usage of intensity scaling to DSC of 0.101. On the other hand, U-Net with resnet18 that employs intensity scaling achieved the lowest HD of 4.071.

SAM with Simulated Human Interaction SAM with SHI achieves the highest DSC of 0.704 using ViT-Huge backbone with bounding box x=5 on the small-scale tumor test set. Also, the same model configuration achieves the lowest HD of 1.925. These are reported in Table 5.8.

SAM without Simulated Human Interaction The SAM without SHI using ViT-Huge backbone achieves the highest DSC of 0.051, and the lowest HD was 2.893 using

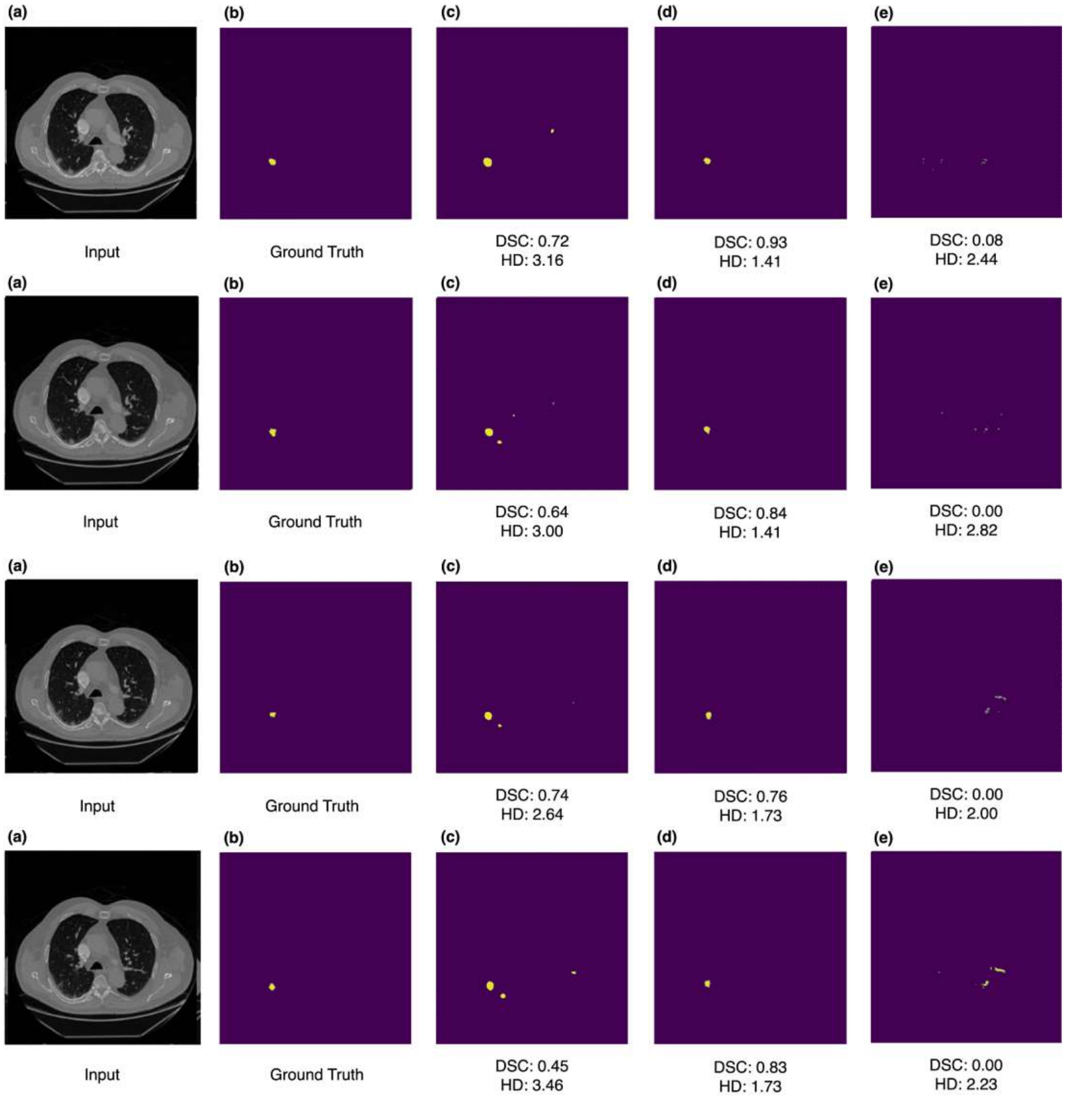


Figure 5.4: Qualitative Results of Lung Cancer Segmentation D_2 Dataset Test Set. The input image **(a)** and ground truth mask **(b)** represent an image pair $p_j^i = \{I_j^i, M_j^i\}$ from dataset D_2. Images **(c)**, **(d)**, and **(e)** represent the output from the best performing CNN-based model (UNet/resnet101), SAM with SHI (SAM/5/Base) and SAM without SHI (SAMN/Huge), respectively.

ViT-Base backbone (Table 5.9).

There is a boxplot provided for the top two models based on DSC for each model architecture in Figure 5.5.

In summary, the increase of DSC and decrease in HD as tumor scale increases indicates that all models struggle with smaller targets. A similar trend in performance between

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.306	0.879	0.342	4.757	6.722	2.125
U-Net	efficientnet-b7	100	0.288	0.868	0.337	4.744	5.570	1.933
U-Net	resnet18	100	0.244	0.837	0.312	4.874	5.828	1.897
U-Net++	efficientnet-b7	100	**0.327**	**0.886**	**0.366**	4.122	4.895	1.621
FPN	resnet18	100	0.223	0.847	0.319	4.145	4.824	1.340
PSPNet	efficientnet-b7	100	0.197	0.776	0.275	4.513	5.800	1.532
U-Net++	resnet152	100	0.273	0.831	0.339	5.294	8.049	2.679
†U-Net++	resnet152	100	0.132	0.606	0.207	6.737	8.594	2.686
U-Net	densenet121	100	0.175	0.687	0.234	6.068	9.304	2.773
††U-Net	resnet18	100	0.101	0.690	0.224	**4.071**	**2.922**	**0.994**
U-Net++	efficientnet-b7	200	0.322	0.875	0.337	4.428	5.170	1.674
U-Net	resnet101	100	0.309	0.828	0.323	4.709	6.022	1.916

Table 5.7: CNN Model Performance on Small Scale Lung Tumor Segmentation Dataset D_2 Test Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	0.694	0.684	0.196	1.969	1.793	0.645
sam-vit-large	5	0.690	0.715	0.219	1.966	1.771	0.579
sam-vit-huge	5	**0.704**	**0.743**	**0.225**	**1.925**	**2.207**	**0.629**
sam-vit-base	20	0.621	0.871	0.286	2.040	1.902	0.612
sam-vit-large	20	0.550	0.824	0.274	2.209	2.071	0.725
sam-vit-huge	20	0.564	0.855	0.280	2.266	2.389	0.761

Table 5.8: SAM with SHI Performance on Small Scale Lung Cancer Segmentation Dataset D_2 Test Set.

tumor scales can be seen in the state-of-the-art paper by Primakov et. al [PIVT$^+$22a]. However, their results remain more consistent based on DSC, but they measure the tumor scales in milliliters (mLs) and perform volumetric DSC evaluation.

5.3.2 Medium Scale Lung Cancer

CNN-based Models The U-Net with resnet101 reaches the highest DSC of 0.601 on the test set (Table 5.10). The HD values are similar for medium-scale tumors compared to small-scale tumors with a HD absolute mean difference of 0.559. Also, DSC results for medium-scale tumors from Table 5.10 are better compared to small-scale tumors from Table 5.7 with an absolute mean difference of 0.195.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.038	0.336	0.152	**2.893**	**1.540**	**0.477**
sam-vit-large	0.039	0.205	0.137	3.131	2.826	0.818
sam-vit-huge	**0.051**	**0.455**	**0.141**	3.000	2.141	0.656

Table 5.9: SAM without SHI Performance on Small Scale Lung Cancer Segmentation Dataset D_2 Test Set.

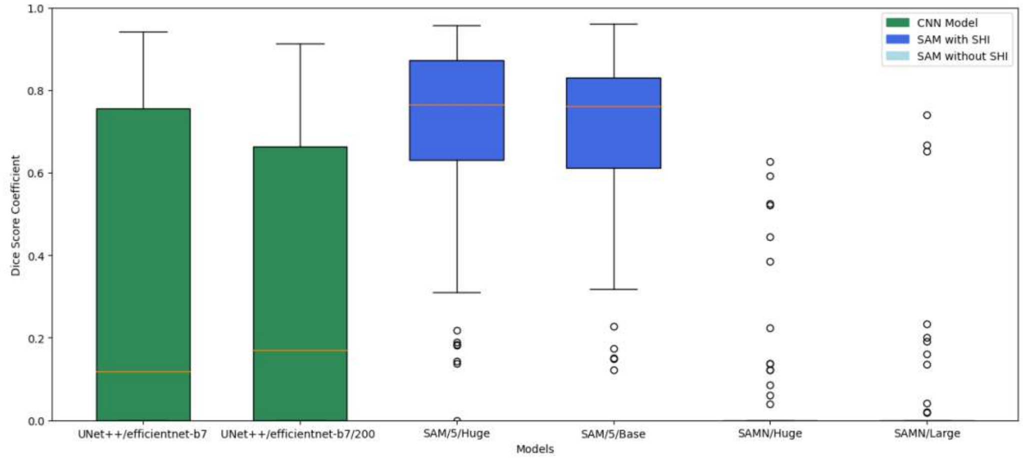


Figure 5.5: Boxplot of Two Best Performing Models on Small Scale Lung Cancer Segmentation Dataset D_2 Test Set Based on DSC. The orange line indicates the median, and the whiskers extend DSC values from minimum to maximum.

SAM with Simulated Human Interaction The highest DSC of 0.804 was reported using SAM with SHI using ViT-Base backbone with x=5 bounding box size, also the lowest HD 2.212 was achieved using the same model (Table 5.11). This is an improvement compared to the best-performing SAM with SHI model applied on a small-scale tumor dataset with an absolute difference in DSC of 0.100.

SAM without Simulated Human Interaction On the medium-scale tumor test set, SAM without SHI using ViT-Huge backbone achieved the best performance based on DSC and HD, with the highest DSC of 0.140 and the lowest HD of 3.662 (Table 5.12).

Similarly, as for small-scale tumors, we include a boxplot in Figure 5.6.

5.3.3 Large Scale Lung Cancer

CNN-based Models The DSC increases drastically for CNN-based models applied on large-scale tumors compared to results for small-scale and medium-scale tumors with DSC absolute mean difference of 0.351 and 0.1566, respectively. U-Net with resnet18 and 100 epochs on the test set achieved the best DSC of 0.771 (Table 5.13). Despite

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.581	0.649	0.225	4.362	3.954	1.321
U-Net	efficientnet-b7	100	0.480	0.892	0.350	4.639	4.627	1.640
U-Net	resnet18	100	0.502	0.902	0.315	4.515	3.993	1.411
U-Net++	efficientnet-b7	100	0.418	0.811	0.298	4.801	3.364	1.181
FPN	resnet18	100	0.389	0.845	0.315	4.616	3.473	1.311
PSPNet	efficientnet-b7	100	0.378	0.773	0.271	4.828	3.341	1.032
U-Net++	resnet152	100	0.539	0.870	0.290	4.510	4.960	1.482
†U-Net++	resnet152	100	0.260	0.644	0.252	6.712	8.325	2.633
U-Net	densenet121	100	0.520	0.771	0.253	4.816	4.657	1.359
††U-Net	resnet18	100	0.322	0.745	0.273	5.088	3.568	1.117
U-Net++	efficientnet-b7	200	0.441	0.773	0.263	4.635	3.303	1.123
U-Net	resnet101	100	**0.601**	**0.860**	**0.271**	**4.307**	**3.805**	**1.340**

Table 5.10: CNN Model Performance on Medium Scale Lung Tumor Segmentation Dataset D_2 Test Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.804**	**0.338**	**0.110**	**2.212**	**1.068**	**0.386**
sam-vit-large	5	0.765	0.461	0.144	2.242	1.267	0.419
sam-vit-huge	5	0.779	0.556	0.160	2.265	1.430	0.457
sam-vit-base	20	0.680	0.591	0.193	2.558	1.852	0.581
sam-vit-large	20	0.692	0.644	0.207	2.476	1.584	0.545
sam-vit-huge	20	0.643	0.876	0.291	2.564	2.009	0.699

Table 5.11: SAM with SHI Performance on Medium Scale Lung Cancer Segmentation Dataset D_2 Test Set.

the overall DSC increase, the HD values remain relatively similar to small-scale tumor results in Table 5.7. Some models even show increased HD (worse performance) on the test set compared to general results in Table 5.4, such as U-Net with 200 epochs.

SAM with Simulated Human Interaction The best-performing model based on both DSC and HD was SAM with SHI using ViT-Base backbone with x=5 bounding box size. It achieved a DSC of 0.868 and HD of 2.293 (Table 5.14).

SAM without Simulated Human Interaction We report results for large-scale tumors using SAM without SHI in Table 5.15. SAM using ViT-Huge backbone achieved the highest DSC of 0.336 and the lowest HD of 3.481.

Lastly, we provide a boxplot for the best-performing models based on DSC for CNN-based and SAM-based models in Figure 5.7.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.082	0.435	0.140	3.930	2.230	0.697
sam-vit-large	0.038	0.314	0.101	4.006	1.898	0.667
sam-vit-huge	**0.140**	**0.600**	**0.220**	**3.662**	**2.103**	**0.671**

Table 5.12: SAM without SHI Performance on Medium Scale Lung Cancer Segmentation Dataset D_2 Test Set.

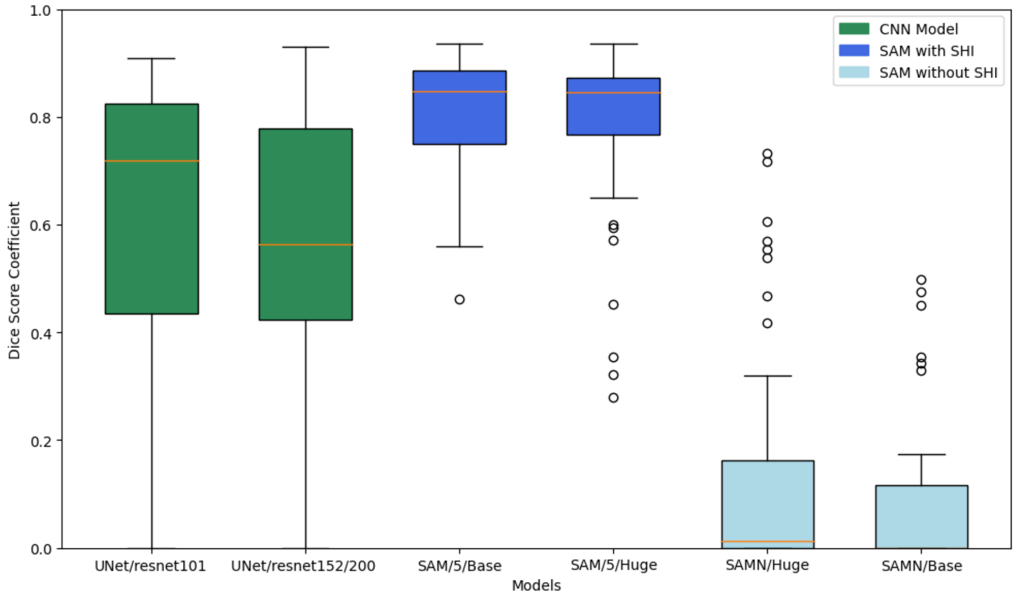


Figure 5.6: Boxplot of Two Best Performing Models on Medium Scale Lung Cancer Segmentation Dataset D_2 Test Set Based on DSC. The orange line indicates the median, and the whiskers extend DSC values from minimum to maximum.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.718	0.433	0.163	4.930	4.239	1.664
U-Net	efficientnet-b7	100	0.725	0.421	0.163	5.058	4.479	1.807
U-Net	resnet18	100	**0.771**	**0.324**	**0.114**	4.269	3.116	1.113
U-Net++	efficientnet-b7	100	0.707	0.341	0.123	**4.219**	**3.051**	**1.131**
FPN	resnet18	100	0.697	0.333	0.128	4.232	3.278	1.246
PSPNet	efficientnet-b7	100	0.591	0.378	0.123	4.550	2.749	1.034
U-Net++	resnet152	100	0.759	0.272	0.094	4.431	3.248	1.232
†U-Net++	resnet152	100	0.300	0.696	0.296	7.913	7.317	2.566
U-Net	densenet121	100	0.720	0.290	0.109	4.792	2.166	0.806
††U-Net	resnet18	100	0.198	0.589	0.229	5.806	3.637	1.235
U-Net++	efficientnet-b7	200	0.685	0.324	0.110	4.410	2.754	1.044
U-Net	resnet101	100	0.756	0.343	0.134	4.925	4.056	1.537

Table 5.13: CNN Model Performance on Large Scale Lung Tumor Segmentation Dataset D_2 Test Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.868**	**0.130**	**0.045**	**2.293**	**1.096**	**0.338**
sam-vit-large	5	0.854	0.127	0.039	2.333	0.691	0.235
sam-vit-huge	5	0.857	0.159	0.052	2.487	1.229	0.434
sam-vit-base	20	0.763	0.223	0.074	2.561	0.902	0.352
sam-vit-large	20	0.756	0.315	0.106	2.488	1.079	0.348
sam-vit-huge	20	0.760	0.280	0.091	2.508	1.267	0.455

Table 5.14: SAM with SHI Performance on Large Scale Lung Cancer Segmentation Dataset D_2 Test Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.047	0.204	0.094	4.643	1.851	0.658
sam-vit-large	0.235	0.450	0.149	4.131	2.284	0.699
sam-vit-huge	**0.336**	**0.738**	**0.229**	**3.481**	**2.582**	**0.792**

Table 5.15: SAM without SHI Performance on Large Scale Lung Cancer Segmentation Dataset D_2 Test Set.

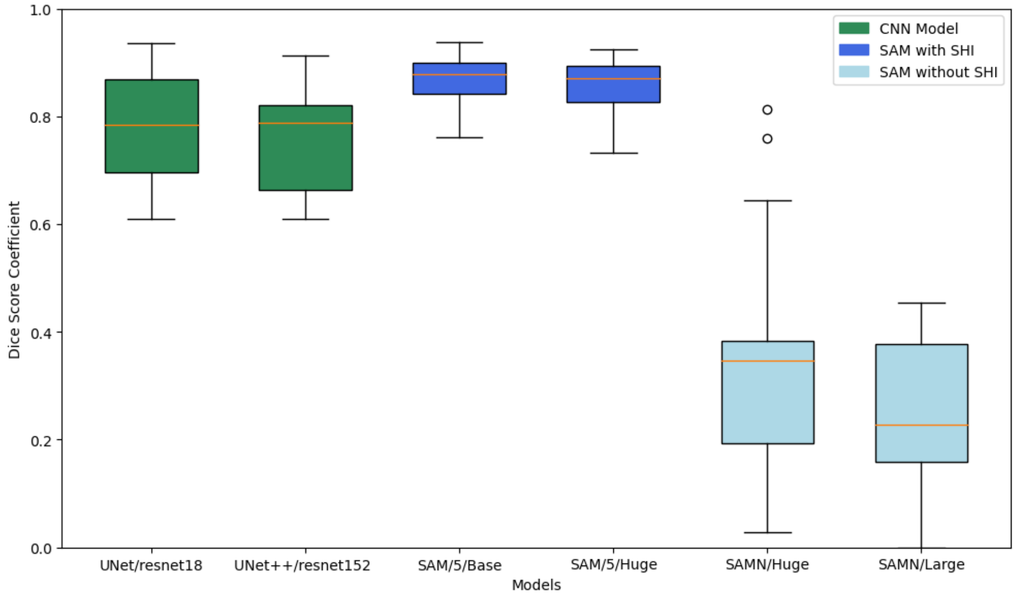


Figure 5.7: Boxplot of Two Best Performing Models on Large Scale Lung Cancer Segmentation Dataset D_2 Test Set Based on DSC. The orange line indicates the median, and the whiskers extend DSC values from minimum to maximum.

CHAPTER 6

Discussion and Conclusion

This chapter highlights the main insights from the thesis in Section 6.1. Contributions are listed in Section 6.2. Subsequently, we describe limitations in Section 6.3. Finally, future work and conclusions are provided in Section 6.4.

6.1 Key Findings

The research questions originally presented in Chapter 1 are revisited in this section, with a focus on key discoveries from the results.

> *How does the CNN-based and ViT-based model performance vary across different datasets?*

Although we attempted to design a robust model that achieves similar performance on both datasets D_1 and D_2, there is still a significant gap between segmenting small and larger targets in terms of task complexity according to Antonelli et al. [ARB+22]. Therefore, the same model may perform worse for highly complex targets such as NSCLC compared to lung targets.

In the case of CNN-based architectures, as outlined in Table 5.1 and Table 5.4, the highest DSC was 0.975 using U-Net with efficientnet-b7 on D_1 dataset, compared to 0.440 using U-Net with resnet101 on D_2 dataset, thereby demonstrating a significant performance variation between the datasets. Consistently across all evaluated models, we observed this disparity in performance when comparing the results of the lung segmentation task D_1 to the lung cancer segmentation task D_2, with the latter yielding lower DSC scores. We hypothesize that this is related to two factors: target size and complexity of the task. Regarding the former, it is known that the size of the target lesion may have a substantial effect on the metric value [RET+21]. In particular, a smaller structure will more likely lead to a lower DSC due to its definition [RET+21]. Regarding the task complexity, segmenting the lung is a significantly easier task compared to detecting and segmenting lung cancer. Lung cancer detection and segmentation involves precise localization of the lesion which can be in different shapes and appearances. In addition, this task entails differentiation between lesions and non-pathological structures that are visually similar. Regarding the lung cancer segmentation D_2 dataset, it is important to note that these results cannot be directly compared to Primakov et al. [PIVT+22a] because we were measuring DSC and HD for each slice, whereas they use 3D CNNs with volumetric metrics.

We achieved the lowest HD distance score of 5.859 and 5.749 with CNN-based and SAM without SHI, respectively. These findings indicate minimal difference between these two models. We conjecture that the larger target size contributes to stabilizing the HD score.

SAM with SHI achieves the highest DSC of 0.966 on D_1, which is lower compared to the CNN-based model despite utilizing the ground truth with SHI which even makes it an unfair comparison. The reason for this may be related to bounding box shape which is not ideal since we have two separate targets with background in-between. This is reflected in the qualitative results, especially in the 4th row of Figure 5.2, where SAM with SHI segments background in-between lungs as a target. However, SAM with SHI has almost 2× improvement on D_2 dataset (DSC of 0.749), compared to the best performing CNN-based model, indicating that the bounding box around the tumor simplifies the segmentation task (Table 5.2 and Table 5.5). We hypothesize that a bounding box around a small target tackles the issue of distinguishing between lesion and non-pathological structures which eases the segmentation task since we provide a preliminary ROI to the model. It is known that bounding box prompts and fine-tuning SAM for specific targets increases the SAM performance.

As reported in Table 5.3 and Table 5.6, SAM without SHI achieves a maximum DSC of 0.920 on D_1, which is still a competitive result compared to CNN-based models. SAM without SHI only achieves a DSC of 0.113 in D_2 (lung cancer segmentation). This is reflected in boxplots, where we depict the two best performing CNN-based, SAM with SHI, and SAM without SHI models for dataset D_1 and D_2 (Figure 5.1, Figure 5.3). Unlike SAM with SHI which utilizes a ground truth for SHI, SAM without SHI offers a fair comparison to previously mentioned CNN-based models because ground truth is not used as input during inference. This indicates that ViT-based model SAM is worse compared to CNN-based models, especially when it comes to small targets like NSCLC, aligning with the conclusion of Huang et al. [HYL+24]. This may be partly explained by the fact that SAM was not supposed to be used without any input or prompt [KMR+23].

> *What is the impact of simulated human interaction with SAM in terms of performance?*

A performance impact with SHI is relatively limited on dataset D_1, SAM with SHI achieves 0.966 as the highest DSC compared to 0.920 using SAM without SHI (Table 5.2 and Table 5.3). In contrast, when it comes to highly imbalanced dataset D_2, SAM with SHI performs significantly better compared to SAM without SHI. This is reported in 5.5 and Table 5.6, where the best SAM with SHI and without SHI achieve a DSC of

0.749 and 0.113, respectively. Regarding lung segmentation dataset D_1, simulated human interaction does not have such an impact because the lung is considered a larger target. We assume that the entire image can be seen as a bounding box when it comes to lung segmentation target due to its size, therefore we have results similar to SAM with SHI. This is particularly reflected in the qualitative results where the model slightly over-segments the target, but it keeps up with the lung structure (Figure 5.2). However, SAM without SHI does not output any reasonable prediction for each example in qualitative results (Figure 5.4). In addition, this aligns with other studies that show that SAM fails in situations with smaller targets, low contrast, irregular shapes, or weak boundaries [ZSJ24].

> *How does the CNN-based and ViT-based model performance compare across different tumor scales?*

To further enrich the model's comparison, we conducted an evaluation stratified by tumor scale explained in Section 4.7, ranging from small to large tumor size. Regarding the small-scale tumor, the highest DSC was 0.327 using U-Net++ with efficientnet-b7, 0.704 using SAM with SHI, and 0.051 using SAM without SHI. This small-scale tumor set can be seen as the most challenging task to segment due to extremely small targets, thus it is reasonable to receive such low numbers, especially for SAM without SHI (Table 5.7, Table 5.8, Table 5.9). However, SAM with SHI increases DSC for small-scale tumors since the bounding box scales the size of the target which simplifies the segmentation task for the SAM model. This is visible in qualitative results, where we compare two best-performing models based on architectures with boxplot (Figure 5.5).

The performance across all CNN-based models improved on medium-scale tumors, the DSC results range from 0.260 (U-Net++/resnet152) to 0.601 (U-Net/resnet101), which is almost 2× an improvement compared to the small-scale best performance model (Table 5.10). The SAM with SHI and without SHI achieve smaller increase compared to CNN-based models, where the highest DSC is 0.804 and 0.140, respectively (Table 5.11, Table 5.12). We hypothesize that the DSC improvement on CNN-based and SAM-based models is related to the larger target compared to small-scale tumors. This aligns with Reinke et al. [RET+21] conclusion, where the larger structures achieve the higher DSC. We provide qualitative results for two best-performing models on a medium-scale dataset (Figure 5.6).

U-Net with resnet18 model evaluated on large-scale tumors achieved the highest DSC of 0.771, which is by far the best performance for the lung cancer segmentation task with CNN-based models (Table 5.13). The model with the same configuration using intensity scaling, U-Net with resnet18, achieved the worst result of only 0.198 DSC on large-scale tumors. SAM with SHI and without SHI achieved DSC of 0.868 and 0.336, respectively (Table 5.14, Table 5.15). Similarly, as for medium-scale models, we assume that the larger target simplifies the segmentation task and increases the DSC [RET+21]. We provide visualization of these results in Figure 5.7. Finally, we achieved the best results with large-scale tumor datasets for each model configuration compared to previous models on either small-scale, medium-scale, or all-scale tumors. It has been evident that SAM achieved a higher performance towards larger objects according to Mazurowski et al. [MDG+23b]. Depending on the tumor size different CNN-based model configurations seem to be optimal.

> *What are appropriate strategies to optimize the performance of deep learning models for the segmentation of CT scans?*

In the preprocessing phase, we used data augmentation, which did not increase nor stabilize the CNN-based model's results. In fact, U-Net++ with resnet152 was one of the best-performing models that achieved a DSC of 0.400. However, the same configuration using data augmentation achieved only a DSC of 0.234, which was the 2nd worst result across all CNN-based models. This was the opposite effect compared to other studies where the robustness and accuracy of the model improved with data augmentation ([RGC+21], [LFSM24], [SKK+23]). We hypothesize that the lung cancer dataset D_2 lacked a sufficient number of CT scans, and hence, increasing the dataset size is crucial for the effectiveness of this method [PIVT+22a].

In addition, data augmentation increased the training time significantly from ∼8 hours without data augmentation to ∼24 hours. We hypothesize that the longer time until convergence is due to the higher variability between each image's target. For example, target size may vary significantly compared to training without data augmentation that uses scaling. However, this makes it harder for a model to learn the features that identify the tumors with smaller sizes.

Also, intensity scaling (U-Net/resnet18) did not contribute to improving the model's performance since it had the worst DSC of only 0.178 compared to all other models, as reported in Table 5.4. Therefore, we did not proceed with using data augmentation and intensity scaling for SAM models. However, in the study where they used PET/CT scans with the DynUNet model for tumor segmentation, intensity scaling increased the DSC [HMS23]. The reason for this discrepancy might be related to the DynUNet model, which unlike the original U-Net uses strided convolutions instead of max-pooling for downsampling.

Experimenting with different model architectures and configurations led to a clear increase in performance, especially for dataset D_2. In CNN-based models, this difference can be seen with PSPNet using efficientnet-b7 which achieves only 0.289 DSC compared to U-Net using resnet101 with 0.440 DSC. The SD is very high for all CNN-based models which indicates challenges with particular slices, as reported in Table 5.4.

In dataset D_1, U-Net with efficientnet-b7 significantly outperformed SAM without SHI and a ViT-Base backbone, achieving a DSC of 0.975 compared to SAM's range of 0.831 to 0.920. Unlike SAM without SHI, CNN-based models demonstrated minimal variance in performance, with DSC ranging from 0.970 to 0.975 and minimal SD (Table 5.3 and Table 5.1). CNN-based models outperform SAM models in lung segmentation tasks, indicating that SAM models heavily rely on the bounding box. This difference between

CNN-based and SAM models is less prominent with larger targets compared to smaller targets with weaker boundaries, which aligns with Zhang et al. [ZSJ24] conclusion.

SAM with SHI performed well on both datasets D_1 and D_2. Regarding dataset D_1, the best performing SAM with SHI model was using ViT-Base backbone and BB set to `x=20`, achieving DSC of 0.966 (Table 5.2). The BB set to `x=5` generally performed worse compared to the BB set to `x=20`. The results for dataset D_2 were ranging from 0.612 to 0.749 DSC using SAM with SHI ViT-Base backbone and BB set to `x=5` (Table 5.5). These findings highlight the importance of BB settings, especially when it comes to smaller targets such as lung cancer. For instance, BB set to `x=5` significantly scales up the tumor and simplifies the segmentation task. Also, SD is very low using the best performing model for dataset D_2. CNN-based model, SAM with SHI, and SAM without SHI outputs are visualized in Figure 5.2 and Figure 5.4. Based on Figure 5.2, we observe that SAM without SHI has more noise compared to other models. Also, it is visible that the CNN-based model is over-segmenting on dataset D_2 in Figure 5.4.

6.2 Contribution

The primary contributions of this thesis are outlined as follows:

- The implementation of CNN-based and ViT-based models with a range of configurations, including ViT-based SAM with and without SHI, as referenced in Table 4.3.
- An in-depth analysis of CNN-based and ViT-based model's performance across different tumor scales.
- Investigating the impact of SHI with fine-tuned Segment Anything Model for different tasks including lung segmentation and lung cancer segmentation.
- Conducting direct comparisons between CNN-based and ViT-based models with various configurations for both lung segmentation and lung cancer segmentation datasets.

6.3 Limitations

In this work, we have faced several major obstacles and limitations. Firstly, at the time of this research, SAM was designed only for 2D image segmentation with a positive target that needs to be prompted. This prevented us to implement and explore 3D CNN models since the results would be incomparable. Therefore, our focus was narrowed down to positive 2D slices that include lung segmentation and non-small cell lung cancer segmentation. This approach introduced another limitation, where we could not compute volumetric DSC like Primakov et al. [PIVT$^+$22a], making our outcomes hardly comparable to results from other prominent papers.

Another major obstacle with SAM was its limitation regarding model fine-tuning. Initially, SAM was designed to be a zero-shot model, but we used an "unofficial" way to fine-tune it for this task [Rog23] [Bha23]. Since SAM was published only very recently (05.04.2023), there was a lot of development done in parallel to our work. For example, MedSAM from Ma et al. [MHL$^+$23] was not utilized since it was published after the coding phase of this thesis was finalized.

Finally, larger datasets are crucial for NSCLC segmentation, given that pre-trained backbones are unfamiliar with this type of complex target. For example, Primakov et al. [PIVT$^+$22a] collected 10 datasets for lung cancer segmentation, which included over 60.000 unique slices. Although comparisons between 2D and 3D volumetric DSC are not straightforward, our DSC results were lower on the NSCLC dataset D_2. However, we achieved state-of-the-art results on dataset D_1, likely because the target was larger.

6.4 Future Work and Conclusion

This thesis investigates various deep learning models, particularly CNN-based and SAM models in the context of lung segmentation and lung cancer segmentation which are important tasks in clinical treatments. Additionally, we investigate the impact of simulated human interaction with the Segment Anything Model.

Our comprehensive analysis revealed differences in performance based on DSC and HD between CNN-based, SAM with SHI, and SAM without SHI, especially with small targets such as NSCLC. When it comes to the lung segmentation task, CNN-based models demonstrated superior performance (Table 5.1). However, the CNN-based model's performance dropped significantly on the lung cancer segmentation task (Table 5.4). SAM with SHI using ViT-Base backbone substantially improved the overall score (Table 5.5). These findings demonstrated the significant positive impact of prompt-based interaction, especially when it comes to smaller and more complex targets.

Since SAM with SHI demonstrated excellent performance on NSCLC dataset D_2, it opens a possibility for radiologists to utilize software in which they can roughly annotate detected tumors, which will be then precisely measured by SAM. Except that, it would be interesting to see the performance of a more complete processing pipeline that combines the detection and segmentation models for NSCLC. In this approach, we would be able to compute volumetric DSC that would be comparable to other state-of-the-art papers like Primakov et al. [PIVT$^+$22a]. Also, an implementation of 3D CNN-based models would be valuable, especially comparing them to ViT-based 3D segmentation models.

It would be particularly interesting to combine different CNN-based model configurations for different tumor scales. According to our results, U-Net with resnet18 would be ideal for larger tumor targets, while for smaller targets, we could utilize U-Net++ with efficientnet-b7.

Expanding the dataset is key for further research on NSCLC segmentation because it will make our models more robust according to Primakov et al. [PIVT$^+$22a], where they combined 10 NSCLC datasets (only some were publicly available). Ideally, we would have a fully automated pipeline for lung segmentation and NSCLC segmentation, similar to Primakov et al. [PIVT$^+$22a] with initial lung isolation, which combines strengths of CNN-based, SAM with SHI, and SAM without SHI models that are evident from our results.

In conclusion, this thesis shows promising capabilities of CNN-based, SAM with SHI, and SAM without SHI for different segmentation tasks. However, further research is needed that involves combining different model configurations and increasing the dataset size, especially for the lung cancer segmentation task.

CHAPTER 7

Appendix

We report results for lung segmentation dataset D_1, lung cancer segmentation dataset D_2, small-scale lung cancer segmentation, medium-scale lung cancer segmentation, and large-scale lung cancer segmentation for CNN-based, SAM with SHI, and SAM without SHI models. These results include only validation sets. They cover configuration details, and evaluation metrics including mean DSC and HD. Also, we report H95th and SD. The highest DSC and lowest HD are highlighted for each table.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	efficientnet-b7	50	0.979	0.024	0.008	6.149	3.294	1.139
U-Net	resnet18	50	0.980	0.023	0.007	**5.976**	**2.397**	**0.817**
U-Net	resnet34	50	0.980	0.021	0.007	6.108	2.737	0.894
U-Net	resnet152	50	**0.981**	**0.025**	**0.008**	5.999	3.464	1.094
U-Net	resnet152	200	0.980	0.025	0.009	6.068	3.334	1.065

Table 7.1: CNN Model Performance on Lung Segmentation Dataset D_1 Validation Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	0.968	0.053	0.020	**4.598**	**2.367**	**0.786**
sam-vit-large	5	0.966	0.066	0.022	4.828	2.306	0.753
sam-vit-huge	5	0.968	0.043	0.019	4.660	2.492	0.825
sam-vit-base	20	**0.968**	**0.035**	**0.019**	4.713	2.060	0.657
sam-vit-large	20	0.966	0.058	0.021	4.828	2.157	0.806
sam-vit-huge	20	0.966	0.053	0.021	4.743	2.387	0.885

Table 7.2: SAM with SHI Performance on Lung Segmentation Dataset D_1 Validation Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.801	0.416	0.129	7.731	3.897	1.198
sam-vit-large	0.906	0.184	0.056	6.899	3.728	1.341
sam-vit-huge	**0.922**	**0.139**	**0.052**	**5.729**	**3.614**	**1.141**

Table 7.3: SAM without SHI Performance on Lung Segmentation Dataset D_1 Validation Set.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.649	0.897	0.276	3.961	3.550	1.068
U-Net	efficientnet-b7	100	0.697	0.912	0.280	3.846	3.295	1.016
U-Net	resnet18	100	0.694	0.557	0.202	3.860	3.563	1.123
U-Net++	efficientnet-b7	100	**0.745**	**0.519**	**0.184**	**3.601**	**3.557**	**1.102**
FPN	resnet18	100	0.674	0.899	0.275	3.821	3.516	1.078
PSPNet	efficientnet-b7	100	0.698	0.903	0.243	3.788	3.678	1.163
U-Net++	resnet152	100	0.679	0.914	0.239	3.875	3.466	1.041
*U-Net++	resnet152	100	0.361	0.864	0.334	6.046	9.010	3.119
U-Net	densenet121	100	0.562	0.889	0.314	4.365	3.277	1.022
**U-Net	resnet18	100	0.397	0.783	0.295	4.786	4.721	1.355
U-Net++	efficientnet-b7	200	0.727	0.444	0.167	3.804	3.521	1.134
U-Net	resnet101	100	0.666	0.890	0.261	3.843	3.240	1.035

Table 7.4: CNN Model Performance on Lung Cancer Segmentation Dataset D_2 Validation Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.846**	**0.195**	**0.065**	**2.282**	**2.049**	**0.630**
sam-vit-large	5	0.827	0.246	0.085	2.337	2.191	0.626
sam-vit-huge	5	0.792	0.433	0.126	2.426	2.071	0.627
sam-vit-base	20	0.781	0.457	0.178	2.475	2.828	0.798
sam-vit-large	20	0.722	0.918	0.247	2.595	2.438	0.756
sam-vit-huge	20	0.715	0.605	0.204	2.596	2.267	0.688

Table 7.5: SAM with SHI Performance on Lung Cancer Segmentation Dataset D_2 Validation Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.186	0.852	0.326	3.703	3.367	1.150
sam-vit-large	**0.217**	**0.746**	**0.274**	**3.658**	**2.747**	**0.879**
sam-vit-huge	0.158	0.810	0.303	3.818	3.353	1.157

Table 7.6: SAM without SHI Performance on Lung Cancer Segmentation Dataset D_2 Validation Set.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.387	0.853	0.342	3.782	3.046	0.944
U-Net	efficientnet-b7	100	0.427	0.865	0.364	3.494	2.455	0.759
U-Net	resnet18	100	0.586	0.864	0.292	3.204	2.443	0.786
U-Net++	efficientnet-b7	100	**0.633**	**0.869**	**0.249**	**3.001**	**2.117**	**0.685**
FPN	resnet18	100	0.404	0.816	0.326	3.493	2.326	0.750
PSPNet	efficientnet-b7	100	0.511	0.824	0.317	3.135	2.236	0.692
U-Net++	resnet152	100	0.508	0.802	0.301	3.283	0.491	0.157
*U-Net++	resnet152	100	0.158	0.781	0.256	6.573	11.535	4.356
U-Net	densenet121	100	0.227	0.746	0.261	4.275	2.490	0.945
**U-Net	resnet18	100	0.209	0.610	0.253	3.936	2.363	0.808
U-Net++	efficientnet-b7	200	0.622	0.865	0.235	3.034	1.873	0.643
U-Net	resnet101	100	0.417	0.823	0.323	3.395	1.876	0.670

Table 7.7: CNN Model Performance on Small Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.802**	**0.193**	**0.064**	**1.773**	**1.035**	**0.349**
sam-vit-large	5	0.779	0.235	0.077	1.824	0.821	0.295
sam-vit-huge	5	0.706	0.487	0.159	1.987	1.585	0.470
sam-vit-base	20	0.705	0.7994	0.227	2.012	1.585	0.541
sam-vit-large	20	0.533	0.847	0.319	2.321	2.191	0.733
sam-vit-huge	20	0.590	0.840	0.273	2.300	2.049	0.626

Table 7.8: SAM with SHI Performance on Small Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	**0.053**	**0.645**	**0.187**	**2.864**	**1.369**	**0.397**
sam-vit-large	0.014	0.080	0.041	3.221	1.909	0.595
sam-vit-huge	0.013	0.043	0.056	2.871	1.369	0.411

Table 7.9: SAM without SHI Performance on Small Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.744	0.454	0.127	4.085	3.921	1.196
U-Net	efficientnet-b7	100	**0.799**	**0.301**	**0.096**	4.021	3.563	1.136
U-Net	resnet18	100	0.718	0.361	0.118	4.252	3.835	1.180
U-Net++	efficientnet-b7	100	0.782	0.377	0.116	**3.900**	**4.065**	**1.229**
FPN	resnet18	100	0.776	0.473	0.136	4.000	3.921	1.247
PSPNet	efficientnet-b7	100	0.754	0.464	0.142	4.104	4.270	1.292
U-Net++	resnet152	100	0.727	0.491	0.157	4.154	3.796	1.126
*U-Net++	resnet152	100	0.377	0.839	0.310	6.531	9.589	3.160
U-Net	densenet121	100	0.673	0.707	0.206	4.510	3.699	1.112
**U-Net	resnet18	100	0.496	0.790	0.276	4.934	3.698	1.222
U-Net++	efficientnet-b7	200	0.769	0.325	0.095	4.008	3.954	1.187
U-Net	resnet101	100	0.756	0.464	0.121	4.088	3.592	1.179

Table 7.10: CNN Model Performance on Medium Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.846**	**0.174**	**0.055**	**2.465**	**2.191**	**0.654**
sam-vit-large	5	0.826	0.241	0.083	2.539	2.140	0.673
sam-vit-huge	5	0.810	0.269	0.094	2.583	1.873	0.660
sam-vit-base	20	0.783	0.442	0.153	2.682	2.850	0.918
sam-vit-large	20	0.772	0.321	0.165	2.753	2.391	0.818
sam-vit-huge	20	0.733	0.396	0.133	2.727	2.391	0.758

Table 7.11: SAM with SHI Performance on Medium Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.317	0.871	0.387	3.659	3.338	1.094
sam-vit-large	**0.318**	**0.776**	**0.300**	**3.653**	**2.960**	**0.882**
sam-vit-huge	0.298	0.826	0.369	3.799	3.179	1.014

Table 7.12: SAM without SHI Performance on Medium Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

			DSC			HD		
Model	**Backbone**	**Epochs**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
U-Net	resnet152	200	0.851	0.105	0.039	3.831	1.375	0.557
U-Net	efficientnet-b7	100	0.889	0.072	0.028	3.910	2.300	0.740
U-Net	resnet18	100	0.839	0.106	0.038	**3.690**	**1.845**	**0.704**
U-Net++	efficientnet-b7	100	0.882	0.107	0.052	3.707	1.730	0.650
FPN	resnet18	100	0.862	0.138	0.043	3.810	2.008	0.694
PSPNet	efficientnet-b7	100	**0.891**	**0.066**	**0.024**	3.942	2.021	0.755
U-Net++	resnet152	100	0.871	0.194	0.061	4.048	1.760	0.623
*U-Net++	resnet152	100	0.700	0.872	0.306	5.259	7.111	2.334
U-Net	densenet121	100	0.867	0.130	0.048	3.944	1.386	0.528
**U-Net	resnet18	100	0.409	0.768	0.248	6.134	4.670	1.577
U-Net++	efficientnet-b7	200	0.791	0.205	0.098	4.712	1.946	0.638
U-Net	resnet101	100	0.855	0.110	0.035	3.827	1.662	0.643

Table 7.13: CNN Model Performance on Large Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

		DSC			HD		
Model	**Bounding Box**	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	5	**0.912**	**0.051**	**0.018**	**2.587**	**1.278**	**0.388**
sam-vit-large	5	0.907	0.063	0.019	2.595	1.062	0.332
sam-vit-huge	5	0.878	0.076	0.026	2.690	1.080	0.329
sam-vit-base	20	0.896	0.081	0.028	2.642	1.178	0.367
sam-vit-large	20	0.881	0.125	0.042	2.597	1.062	0.418
sam-vit-huge	20	0.865	0.139	0.081	2.707	1.152	0.361

Table 7.14: SAM with SHI Performance on Large Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

	DSC			HD		
Model	**mean**	**H95th**	**SD**	**mean**	**H95th**	**SD**
sam-vit-base	0.034	0.093	0.032	5.143	1.674	0.560
sam-vit-large	**0.258**	**0.646**	**0.235**	**4.358**	**2.590**	**0.788**
sam-vit-huge	0.003	0.0164	0.007	5.364	1.403	0.473

Table 7.15: SAM without SHI Performance on Large Scale Lung Cancer Segmentation Dataset D_2 Validation Set.

List of Figures

List of Tables

Bibliography

[AHL+20] Haikal Abdulah, Benjamin Huber, Sinan Lal, Hassan Abdallah, Hamid Soltanian-Zadeh, and Domenico L. Gatti. Lung segmentation in chest x-rays with res-cr-net, 2020.

[ARB+22] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Bram van Ginneken, Michel Bilello, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc J. Gollub, Stephan H. Heckers, Henkjan Huisman, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Jennifer S. Golia Pernicka, Kawal Rhode, Catalina Tobon-Gomez, Eugene Vorontsov, James A. Meakin, Sebastien Ourselin, Manuel Wiesenfarth, Pablo Arbeláez, Byeonguk Bae, Sihong Chen, Laura Daza, Jianjiang Feng, Baochun He, Fabian Isensee, Yuanfeng Ji, Fucang Jia, Ildoo Kim, Klaus Maier-Hein, Dorit Merhof, Akshay Pai, Beomhee Park, Mathias Perslev, Ramin Rezaiifar, Oliver Rippel, Ignacio Sarasua, Wei Shen, Jaemin Son, Christian Wachinger, Liansheng Wang, Yan Wang, Yingda Xia, Daguang Xu, Zhanwei Xu, Yefeng Zheng, Amber L. Simpson, Lena Maier-Hein, and M. Jorge Cardoso. The medical segmentation decathlon. *Nature Communications*, 13(1), July 2022.

[ASEHM18] Brahim Ait Skourt, Abdelhamid El Hassani, and Aicha Majda. Lung ct image segmentation using deep neural networks. *Procedia Computer Science*, 127:109–113, 2018.

[aug24] Augmenters contrast — imgaug 0.4.0 documentation. `https://imgaug.readthedocs.io/en/latest/source/overview/contrast.html#linearcontrast`, 2024. Accessed: 2024-03-12.

[BDPW22] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022.

[BFS+18] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L. Siegel, Lindsey A. Torre, and Ahmedin Jemal. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 68(6):394–424, sep 2018.

[Bha23] Sreenivas Bhattiprolu. Fine-tune segment anything model (sam) mito dataset. `https://github.com/bnsreenu/python_for_microscopists/blob/master/331_fine_tune_SAM_mito.ipynb`, 2023. Accessed: Jan. 9, 2024.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BKC16] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[BMR$^+$20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[CC17] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, December 2017.

[Che24] Chest x-ray dataset with lung segmentation v1.0.0. `https://physionet.org/content/chest-x-ray-segmentation/1.0.0/`, 2024. Accessed: 2024-02-25.

[CLB$^+$22] M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Andriy Myronenko, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng Tang, Isaac Yang, Michael Zephyr, Behrooz Hashemian, Sachidanand Alle, Mohammad Zalbagi Darestani, Charlie Budd, Marc Modat, Tom Vercauteren, Guotai Wang, Yiwen Li, Yipeng Hu, Yunguan Fu, Benjamin Gorman, Hans Johnson, Brad Genereaux, Barbaros S. Erdal, Vikash Gupta, Andres Diaz-Pinto, Andre Dourson, Lena Maier-Hein, Paul F. Jaeger, Michael Baumgartner, Jayashree Kalpathy-Cramer, Mona Flores, Justin Kirby, Lee A. D. Cooper, Holger R. Roth, Daguang Xu, David Bericat, Ralf Floca, S. Kevin Zhou, Haris Shuaib, Keyvan Farahani, Klaus H. Maier-Hein, Stephen Aylward, Prerna Dogra, Sebastien Ourselin, and

Andrew Feng. Monai: An open-source framework for deep learning in healthcare, 2022.

[CPSA17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[CT224] Computed tomography (ct). `https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct`, 2024. Accessed: 2024-02-29.

[CVS+13] Kenneth W. Clark, Bruce A. Vendt, Kirk E. Smith, John B. Freymann, Justin S. Kirby, Paul Koppel, Stephen M. Moore, Stanley R. Phillips, David R. Maffitt, Michael Pringle, Lawrence Tarbox, and Fred W. Prior. The cancer imaging archive (tcia): Maintaining and operating a public information repository. *J. Digital Imaging*, 26(6):1045–1057, 2013.

[CWC+23] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 205–218, Cham, 2023. Springer Nature Switzerland.

[CWP+19] Wei Chen, Haifeng Wei, Suting Peng, Jiawei Sun, Xu Qiao, and Boqiang Liu. Hsn: Hybrid segmentation network for small cell lung cancer segmentation. *IEEE Access*, 7:75591–75603, 2019.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[Dic45] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[DSY+22] Luca Deininger, Bernhard Stimpel, Anil Yuce, Samaneh Abbasi-Sureshjani, Simon Schönenberger, Paolo Ocampo, Konstanty Korski, and Fabien Gaire. A comparative study between vision transformers and cnns in digital pathology, 2022.

[Exc19] Stack Exchange. Visualizing matrix convolution. `https://tex.stackexchange.com/questions/522118/visualizing-matrix-convolution`, 2019. Accessed: Jan. 9, 2024.

[FFC$^+$23] Annarita Fanizzi, Federico Fadda, Maria Colomba Comes, Samantha Bove, Annamaria Catino, Erika Di Benedetto, Angelo Milella, Michele Montrone, Annalisa Nardone, Clara Soranno, Alessandro Rizzo, Deniz Can Guven, Domenico Galetta, and Raffaella Massafra. Comparison between vision transformers and convolutional neural networks to predict non-small lung cancer recurrence. *Scientific Reports*, 13(1), November 2023.

[FL19] Hao Fang and Florent Lafarge. Pyramid scene parsing network in 3d: Improving semantic segmentation of point clouds with multi-scale contextual information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 154:246–258, 2019.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GLGL17] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(2):87–93, November 2017.

[GLL19] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[GML20] Gusztáv Gaál, Balázs Maga, and András Lukács. Attention u-net based adversarial architectures for chest x-ray lung segmentation, 2020.

[Gup23] Divam Gupta. Image segmentation keras : Implementation of segnet, fcn, unet, pspnet and other models in keras, 2023.

[GWK$^+$15] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *CoRR*, abs/1512.07108, 2015.

[HBL$^+$23] Sheng He, Rina Bao, Jingpeng Li, Jeffrey Stout, Atle Bjornerud, P. Ellen Grant, and Yangming Ou. Computer-vision benchmark segment-anything model (sam) in medical images: Accuracy in 12 datasets, 2023.

[HCP$^+$23] Michael James Horry, Subrata Chakraborty, Biswajeet Pradhan, Manoranjan Paul, Jing Zhu, Prabal Datta Barua, U. Rajendra Acharya, Fang Chen, and Jianlong Zhou. Full-resolution lung nodule segmentation from chest x-ray images using residual encoder-decoder networks, 2023.

[HCX$^+$21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021.

[HEO22] Emerald U. Henry, Onyeka Emebob, and Conrad Asotie Omonhinmin. Vision transformers in medical imaging: A review, 2022.

[HKR93] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[HLvdMW18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[HMS23] Matthias Hadlich, Zdravko Marinov, and Rainer Stiefelhagen. Autopet challenge 2023: Sliding window-based optimization of u-net, 2023.

[Hou] Hounsfield unit | radiology reference article | radiopaedia.org. https://radiopaedia.org/articles/hounsfield-unit.

[HPP$^+$20] Johannes Hofmanninger, Forian Prayer, Jeanny Pan, Sebastian Röhrich, Helmut Prosch, and Georg Langs. Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. *European Radiology Experimental*, 4(1), August 2020.

[HYL$^+$24] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, Sijing Liu, Haozhe Chi, Xindi Hu, Kejuan Yue, Lei Li, Vicente Grau, Deng-Ping Fan, Fajin Dong, and Dong Ni. Segment anything model for medical images? *Medical Image Analysis*, 92:103061, February 2024.

[HYR$^+$21] Yufan He, Dong Yang, Holger Roth, Can Zhao, and Daguang Xu. Dints: Differentiable neural network topology search for 3d medical image segmentation, 2021.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[Iak19] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[IPK$^+$18] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F. Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, and Klaus H. Maier-Hein. nnu-net: Self-adapting framework for u-net-based medical image segmentation, 2018.

[Jai20] Vidit Jain. Understanding of convolutional neural network (cnn) — deep learning. https://medium.com/analytics-vidhya/introduction-to-semantic-image-segmentation-856cda5e5de8, 2020. Accessed: Jan. 9, 2024.

[JBZ+22] Xi Jia, Joseph Bartlett, Tianyang Zhang, Wenqi Lu, Zhaowen Qiu, and Jinming Duan. U-net vs transformer: Is u-net outdated in medical image registration?, 2022.

[JFR+21] Yeganeh Jalali, Mansoor Fateh, Mohsen Rezvani, Vahid Abolghasemi, and Mohammad Hossein Anisi. Resbcdu-net: A deep learning framework for lung ct image segmentation. *Sensors*, 21(1), 2021.

[JHS+21] Stephanie T. Jünger, Ulrike Cornelia Isabel Hoyer, Diana Schaufler, Kai Roman Laukamp, Lukas Goertz, Frank Thiele, Jan-Peter Grunz, Marc Schlamann, Michael Perkuhn, Christoph Kabbasch, Thorsten Persigehl, Stefan Grau, Jan Borggrefe, Matthias Scheffler, Rahil Shahzad, and Lenhard Pennig. Fully automated <scp>mr</scp> detection and segmentation of brain metastases in non-small cell lung cancer using deep learning. *Journal of Magnetic Resonance Imaging*, 54(5):1608–1622, May 2021.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[KJvdS17] Baris Kayalibay, Grady Jensen, and Patrick van der Smagt. Cnn-based segmentation of medical imaging data, 2017.

[KMR+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

[LAHYB15] Hassan Lemjabbar-Alaoui, Omer UI Hassan, Yi-Wei Yang, and Petra Buchanan. Lung cancer: Biology and treatment options. *Biochimica et Biophysica Acta (BBA) - Reviews on Cancer*, 1856(2):189–210, dec 2015.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, may 2015.

[LCJ+24] Libin Lan, Pengzhou Cai, Lu Jiang, Xiaojuan Liu, Yongmei Li, and Yudong Zhang. Brau-net++: U-shaped hybrid cnn-transformer network for medical image segmentation, 2024.

[LDD+18] Menglu Liu, Junyu Dong, Xinghui Dong, Hui Yu, and Lin Qi. Segmentation of lung nodule in ct images based on mask r-cnn. In *2018 9th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6, 2018.

[LDG+16] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[LFSM24] Chang Liu, Fuxin Fan, Annette Schwarz, and Andreas Maier. Anatomix: Anatomy-aware data augmentation for multi-organ segmentation, 2024.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[LLY+22] Wufeng Liu, Jiaxin Luo, Yan Yang, Wenlian Wang, Junkui Deng, and Liang Yu. Automatic lung segmentation in chest x-ray images using improved u-net. *Scientific Reports*, 12(1), May 2022.

[LWT+21] Tingting Liang, Yongtao Wang, Zhi Tang, Guosheng Hu, and Haibin Ling. Opanas: One-shot path aggregation network architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10195–10203, June 2021.

[LXAW18] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation, 2018.

[Mad17] Scott Mader. Finding and measuring lungs in ct data. `https://www.kaggle.com/datasets/kmader/finding-lungs-in-ct-data/data`, 2017. Accessed: Jan. 7, 2024.

[MDG+23a] Maciej A. Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: An experimental study. *Medical Image Analysis*, 89:102918, 2023.

[MDG+23b] Maciej A. Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: An experimental study. *Medical Image Analysis*, 89:102918, October 2023.

[MHL+23] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images, 2023.

[MHSS21] Christos Matsoukas, Johan Fredin Haslum, Magnus Söderberg, and Kevin Smith. Is it time to replace cnns with transformers for medical images?, 2021.

[Nan23] Georgios Nanos. Neural networks: Pooling layers. `https://www.baeldung.com/cs/neural-networks-pooling-layers`, 2023. Accessed: Jan. 9, 2024.

[NTBA22] S Ali John Naqvi, Abdullah Tauqeer, Rohaib Bhatti, and S Bazil Ali. Improved lung segmentation based on u-net architecture and morphological operations, 2022.

[PGM$^+$19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[PIVT$^+$22a] Sergey Primakov, Abdalla Ibrahim, Janita Van Timmeren, Guangyao Wu, Simon Keek, Manon Beuque, Renée Granzier, Elizaveta Lavrova, Madeleine Scrivener, Sebastian Sanduleanu, Esma Kayan, Iva Halilaj, Anouk Lenaers, Jianlin Wu, René Monshouwer, Xavier Geets, Hester Gietema, Lizza Hendriks, and Philippe Lambin. Automated detection and segmentation of non-small cell lung cancer computed tomography images. *Nature Communications*, 13, 06 2022.

[PIvT$^+$22b] Sergey P. Primakov, Abdalla Ibrahim, Janita E. van Timmeren, Guangyao Wu, Simon A. Keek, Manon Beuque, Renée W. Y. Granzier, Elizaveta Lavrova, Madeleine Scrivener, Sebastian Sanduleanu, Esma Kayan, Iva Halilaj, Anouk Lenaers, Jianlin Wu, René Monshouwer, Xavier Geets, Hester A. Gietema, Lizza E. L. Hendriks, Olivier Morin, Arthur Jochems, Henry C. Woodruff, and Philippe Lambin. Automated detection and segmentation of non-small cell lung cancer computed tomography images. *Nature Communications*, 13(1), jun 2022.

[PKO$^+$17] P.E. Postmus, K.M. Kerr, M. Oudkerk, S. Senan, D.A. Waller, J. Vansteenkiste, C. Escriu, and S. Peters. Early and locally advanced non-small-cell lung cancer (NSCLC): ESMO clinical practice guidelines for diagnosis, treatment and follow-up. *Annals of Oncology*, 28:iv1–iv21, jul 2017.

[PPKS23] Anway S. Pimpalkar, Rashmika K. Patole, Ketaki D. Kamble, and Mahesh H. Shindikar. Performance evaluation of vanilla, residual, and dense 2d u-net architectures for skull stripping of augmented 3d t1-weighted mri head scans, 2023.

[Pre24] Precision and recall - wikipedia. https://en.wikipedia.org/wiki/Precision_and_recall, 2024. Accessed: 2024-03-11.

[Rag18] Prabhu Raghav. Understanding of convolutional neural network (cnn) — deep learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-9976 2018. Accessed: Jan. 9, 2024.

[RET+21] Annika Reinke, Matthias Eisenmann, Minu Dietlinde Tizabi, Carole H. Sudre, Tim Rädsch, Michela Antonelli, Tal Arbel, Spyridon Bakas, M. Jorge Cardoso, Veronika Cheplygina, Keyvan Farahani, Ben Glocker, Doreen Heckmann-Nötzel, Fabian Isensee, Pierre Jannin, Charles Kahn, Jens Kleesiek, Tahsin Kurc, Michal Kozubek, Bennett A. Landman, Geert Litjens, Klaus Maier-Hein, Anne Lousise Martel, bjoern menze, Henning Müller, Jens Petersen, Mauricio Reyes, Nicola Rieke, Bram Stieltjes, Ronald M. Summers, Sotirios A. Tsaftaris, Bram van Ginneken, Annette Kopp-Schneider, Paul Jäger, and Lena Maier-Hein. Common limitations of performance metrics in biomedical image analysis. In *Medical Imaging with Deep Learning*, 2021.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[RGC+21] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness, 2021.

[RHDN23] Stenford Ruvinga, Gordon Hunter, Olga Duran, and Jean-Christophe Nebel. Identifying queenlessness in honeybee hives from audio signals using machine learning. *Electronics*, 12(7):1627, March 2023.

[Rog23] Niels Rogge. Fine-tune sam (segment anything) on a custom dataset. https://github.com/NielsRogge/Transformers-Tutorials/blob/master/SAM/Fine_tune_SAM_(segment_anything)_on_a_custom_dataset.ipynb, 2023. Accessed: Jan. 9, 2024.

[Ros21] Adrian Rosebrock. Backpropagation from scratch with python. https://pyimagesearch.com/2021/05/06/backpropagation-from-scratch-with-python/, 2021. Accessed: Jan. 9, 2024.

[Rud17] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.

[SAB+19] Amber L. Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A. Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M. Summers, Patrick Bilic, Patrick F. Christ, Richard K. G. Do, Marc Gollub, Jennifer Golia-Pernicka, Stephan H. Heckers, William R. Jarnagin, Maureen K. McHugo, Sandy Napel, Eugene Vorontsov, Lena Maier-Hein, and M. Jorge Cardoso. A large annotated medical image dataset for the development and evaluation of segmentation algorithms, 2019.

[SASL23] Yahia Said, Ahmed A. Alsheikhy, Tawfeeq Shawly, and Husam Lahza. Medical images segmentation for lung cancer diagnosis based on deep learning architectures. *Diagnostics*, 13(3):546, feb 2023.

[sch24] File:schematic illustration of ct scanner, extracted from policy implications of the computed tomography (ct) scanner (1978).png - wikimedia commons. https://commons.wikimedia.org/wiki/File:Schematic_illustration_of_CT_scanner,_extracted_from_Policy_implications_of_the_computed_tomography_%28CT%29_scanner_%281978%29.png, 2024. Accessed: 2024-02-29.

[SFW+23] Maximilian Springenberg, Annika Frommholz, Markus Wenzel, Eva Weicken, Jackie Ma, and Nils Strodthoff. From modern cnns to vision transformers: Assessing the performance, robustness, and classification strategies of deep learning models in histopathology. *Medical Image Analysis*, 87:102809, July 2023.

[SGLS21] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7262–7272, October 2021.

[Sim17] Osvaldo Simeone. A brief introduction to machine learning for engineers. 2017.

[SKK+23] Saeko Sasuga, Akira Kudo, Yoshiro Kitamura, Satoshi Iizuka, Edgar Simo-Serra, Atsushi Hamabe, Masayuki Ishii, and Ichiro Takemasa. Image synthesis-based late stage cancer augmentation and semi-supervised segmentation for mri rectal cancer staging, 2023.

[spl24] Spleen segmentation 3d project monai. https://github.com/Project-MONAI/tutorials/blob/main/3d_segmentation/spleen_segmentation_3d.ipynb, 2024. Accessed: 2024-03-12.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[TCERPCU23] Juan Terven, Diana M. Cordova-Esparza, Alfonso Ramirez-Pedraza, and Edgar A. Chavez-Urbiola. Loss functions and metrics in deep learning, 2023.

[TH15a] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2153–2163, 2015.

[TH15b] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1), August 2015.

[TL19] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[TT23] Shweta Tyagi and Sanjay N. Talbar. Predicting lung cancer treatment response from <scp>ct</scp> images using deep learning. *International Journal of Imaging Systems and Technology*, 33(5):1577–1592, April 2023.

[Tur23a] Turing. Understanding feed forward neural networks with maths and statistics. `https://www.turing.com/kb/mathematical-formulation-of-feed-forward-neural-network`, 2023. Accessed: Jan. 9, 2024.

[Tur23b] Turing. What is the necessity of bias in neural networks? `https://www.turing.com/kb/necessity-of-bias-in-neural-networks#how-to-add-bias-to-neural-networks`, 2023. Accessed: Jan. 9, 2024.

[TYL$^+$22] Yucheng Tang, Dong Yang, Wenqi Li, Holger Roth, Bennett Landman, Daguang Xu, Vishwesh Nath, and Ali Hatamizadeh. Self-supervised pre-training of swin transformers for 3d medical image analysis, 2022.

[VGG24] Vgg16 architecture | download scientific diagram. `https://www.researchgate.net/figure/VGG16-architecture_fig1_372072436`, 2024. Accessed: 2024-03-01.

[VSP$^+$23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[Wan20] Wei Wang. Using unet and pspnet to explore the reusability principle of CNN parameters. *CoRR*, abs/2008.03414, 2020.

[WGC$^+$20] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Fast neural architecture search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[WJL+23] Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation, 2023.

[WLX+19] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation, 2019.

[WXL+21] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, 2021.

[WYB+23] Rima Tri Wahyuningrum, Indah Yunita, Achmad Bauravindah, Indah Agustien Siradjuddin, Budi Dwi Satoto, Amillia Kartika Sari, and Anggraini Dwi Sensusiati. Chest x-ray dataset and ground truth for lung segmentation. *Data in Brief*, 51:109640, December 2023.

[WZL+17] Shuo Wang, Mu Zhou, Zaiyi Liu, Zhenyu Liu, Dongsheng Gu, Yali Zang, Di Dong, Olivier Gevaert, and Jie Tian. Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation. *Medical Image Analysis*, 40:172–183, 2017.

[XWW+21] Enze Xie, Wenjia Wang, Wenhai Wang, Peize Sun, Hang Xu, Ding Liang, and Ping Luo. Segmenting transparent object in the wild with transformer, 2021.

[XYZ+19] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Autofpn: Automatic network architecture adaptation for object detection beyond classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[YXZ+23] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. Image data augmentation for deep learning: A survey, 2023.

[ZL23] Kaidong Zhang and Dong Liu. Customized segment anything model for medical image segmentation, 2023.

[ZLZ+21] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, 2021.

[ZSJ24] Yichi Zhang, Zhenrong Shen, and Rushi Jiao. Segment anything model for medical image segmentation: Current applications and future directions, 2024.

[ZSQ+17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017.

[ZSS+19] Zongwei Zhou, Vatsal Sodha, Md Mahfuzur Rahman Siddiquee, Ruibin Feng, Nima Tajbakhsh, Michael B. Gotway, and Jianming Liang. Models genesis: Generic autodidactic models for 3d medical image analysis, 2019.

[ZSTL18] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *CoRR*, abs/1807.10165, 2018.

[ZTT+22] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan liu. Segvit: Semantic segmentation with plain vision transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 4971–4982. Curran Associates, Inc., 2022.

[ZWB+04] Kelly H. Zou, S. Warfield, Aditya Bharatha, Clare M. Tempany, Michael R. Kaus, Steven Haker, William M. Wells, Ferenc A. Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic radiology*, 11 2:178–89, 2004.

[ZZ22] Zhuangzhuang Zhang and Weixiong Zhang. Pyramid medical transformer for medical image segmentation, 2022.