12-2023

LUNG LESION SEGMENTATION USING DEEP LEARNING APPROACHES

Sree Snigdha Tummala

LUNG LESION SEGMENTATION USING DEEP LEARNING APPROACHES

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

by

Sree Snigdha Tummala

December 2023

LUNG LESION SEGMENTATION USING

DEEP LEARNING APPROACHES

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Sree Snigdha Tummala

December 2023

Approved by:

Dr. Jennifer Jin, Advisor, School of Computer Science and Engineering

Dr. Yunfei Hou, Committee Member

Dr. Qingquan Sun, Committee Member

ABSTRACT

The amount of data generated in the medical imaging field, especially in a modern context, is growing significantly. As the amount of data grows, it's prudent to make use of automated techniques that can leverage datasets to solve problems that are error-prone or have inconsistent solutions.

Deep learning approaches have gained traction in medical imaging tasks due to their superior performance with larger datasets and ability to discern the intricate features of 3D volumes, a task inefficient if done manually. Specifically for the task of lung nodule segmentation, several different methods have been tried before such as region growing etc. but this project focuses on using an Attention U-Net model to automatically segment the nodule boundaries. Specifically, this is done on the LUNA16 dataset as a benchmark which is a popular reference point for comparison. To achieve this, specifically, the Attention U-Net was trained with 5-fold cross-validation on the training dataset.

In addition to the segmentation outputs, averaged training and validation curves over all folds were also shown as the model is trained for 70 epochs. To conclude, these results present a useful automated method to segment the lung nodules. In practical situations, this would be of significant help to radiologists as it is less error-prone and not as susceptible to inter-observer variability. These automated tools along with other radiologist interactions could potentially significantly improve patient outcomes.

ACKNOWLEDGEMENTS

I would like to thank Dr. Jennifer Jin, my proposed advisor, for her mentorship and guidance throughout my project. I appreciate the opportunity to experiment on this topic as it is what I'd like to continue my research work in.

I am also thankful to my committee members, Dr. Yunfei Hou and Dr. Qingquan Sun for agreeing to be part of my committee. I hope they find the results to be of interest and I look forward to collaborating with them in future contexts.

I would also like to express my appreciation to my university for offering a conducive learning environment and a wide range of courses that have enabled me to be able to work independently on this topic and also to be able to read the current state-of-the-art research papers.

Last, but certainly not least, I want to extend my thanks to my family and friends for their unending support, encouragement, and belief in my endeavors. Their faith has been a driving force behind my achievements, and I am truly fortunate to have such a strong support system.

I am grateful to each, and every individual mentioned above for their contributions to my academic and personal growth.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Lung lesions, primarily nodules and masses, have long since been seen as potential indicators of diseases, including various malignancies and cancer [1]. Segmenting these data from a patient's CT or MRI data is important in the clinical pipeline aiding in early diagnosis, planning treatment and further monitoring for lung cancer.

Lung cancer is still one of the most prevalent and dangerous cancer types worldwide. Recent statistics show it is a significant part of all cancer-related deaths [2]. Because of this, detecting cancer early is vital is having better patient outcomes [3]. In the past, radiologists have relied on manual or semi-automated and assisted methods for lung lesion detection and delineation. These are time-consuming and subject to inter-observer variability and are thus error-prone [4]. The usage and rapid evolution of deep learning in this field of imaging in the last few years have revolutionized medical imaging analysis. Particularly CNNs have shown state-of-the-art in a wide range of tasks, more specifically computer vision and have outperformed traditional image processing tasks in terms of accuracy and quality of outputs [5]. The potential to use deep learning models in clinical workflows is significant in reducing the amount of time radiologists need to spend on manually delineating the tumors and the exponential growth of datasets available also allows the models to learn features that are generally not immediately visible to the human eye. Furthermore, these models also provide valuable numeric metrics that help in clinical decision-making.

This report shows the application of an advanced deep-learning model for segmenting lung lesions in CT datasets. The subsequent sections go into more detail about the data used and the specific methods.

Challenges

Lung lesion segmentation in CT scans presents a multifaceted challenge. The diverse nature of lesions, coupled with the inherent complexities of lung anatomy and the limitations of imaging technologies, necessitates the development of robust and adaptable methodologies. The following list enumerates some of the prominent challenges faced in the domain:

- Heterogeneity of Lesions: Lung lesions appear in different shapes, sizes and densities. The wide spectrum of lesion appearances, ranging from solid nodules to ground-glass opacities, poses a significant challenge in developing a universally effective segmentation algorithm [6].
- Adjacent Structures: Lesions located near vascular structures, the pleura or other organs are particularly challenging to segment. These can lead to false positives or inaccurate estimation of the lesion's boundary.
- Image Artifacts: CT scans generally contain a lot of artifacts due to motion during acquisition, beam hardening and partial volume effects. These can distort the lesion appearance which complicates the segmentation task [7].

- Low Contrast Lesions: Some lesions, especially early-stage ones might have lower contrast than the surrounding lung. Identifying and segmenting these lesions requires algorithms that can tackle those problems.
- Variability in Imaging Protocols: Different scanning methodologies and protocols such as slice thickness, reconstruction algorithms and radiation doses can result in different image appearances. This causes another problem in the generalization of segmentation models [8].
- Inter-patient Variability: The anatomy and pathology of the lungs can be significantly different between patients. Many factors like patient age, smoking history and other underlying conditions can influence the lesion appearance and require the model to be adaptable to different patient cohorts.

Several published results tackled some of these challenges and will be discussed further in subsequent sections.

CHAPTER TWO

RELATED WORK

Considering the importance of lung lesion segmentation in accurate diagnosis and treatment planning, over the years several different methods have been proposed to tackle this problem starting from simple image processing to sophisticated deep learning methods. This section will cover all the related work that has been published to address this task and further insights into its advantages and limitations.

Image Processing Methods

Historically to improve manual delineation, simple image processing techniques were used. These methods primarily relied on thresholding, morphological operations, region growing and other methods to do lesion segmentation from the surrounding lung.

- Thresholding: This method simply sets the intensity range to segment the lesion from the background. The assumption is that lesions have a different intensity compared to the surroundings. This method was generally effective for high-contrast images with lesions that neatly segment this way but in general, for more complex tasks its performance was subpar [9].

- Morphological Operations: This method used structural elements in the image to identify regions allowing removal of small noise and enhancement of the structures. Like basic thresholding, as the boundary of the lesion gets more complex, it's harder to have a clean segmentation due to more detailed boundaries [10]
- Region Growing: This is a method that starts with a random seed point and tries to grow by detecting similar surrounding pixels. While it has advantages compared to other techniques, this method largely depends on getting the right seed starting point, non-disjointed lesions and having homogenous structures that are easily grown [11].

Machine Learning Methods

With the advent of Machine Learning, feature extraction methods combined with classifiers have become more popular. These methods focus on hand-crafted features that are extracted from the input such as texture, shape and intensity followed by simple classifiers such as SVM, Random Forests and k-NN for segmentation.

- Texture-based features could be extracted with methods such as Gabor filters and co-occurrence matrices [12]
- Classifiers such as SVMs became more used due to their capability to handle high-dimensional datasets [13].

Deep Learning Models

The introduction of deep learning methods has improved the lesion segmentation accuracy significantly. CNNs have become the standard with architectures like U- Net [14] which is a symmetric encoder-decoder architecture as can be seen in Figure 1 emerging as benchmarks for biomedical segmentation in general. The best part of these models is their ability to learn hierarchical features directly from the image data without the need for previously mentioned hand-crafted features. The U-Net model shown will be used as a baseline for comparison of the final method as it is a common comparison. It uses skip connections along with the encoder-decoder architecture. There have been several improvements to this model that improve the performance of segmentation which will be discussed further.

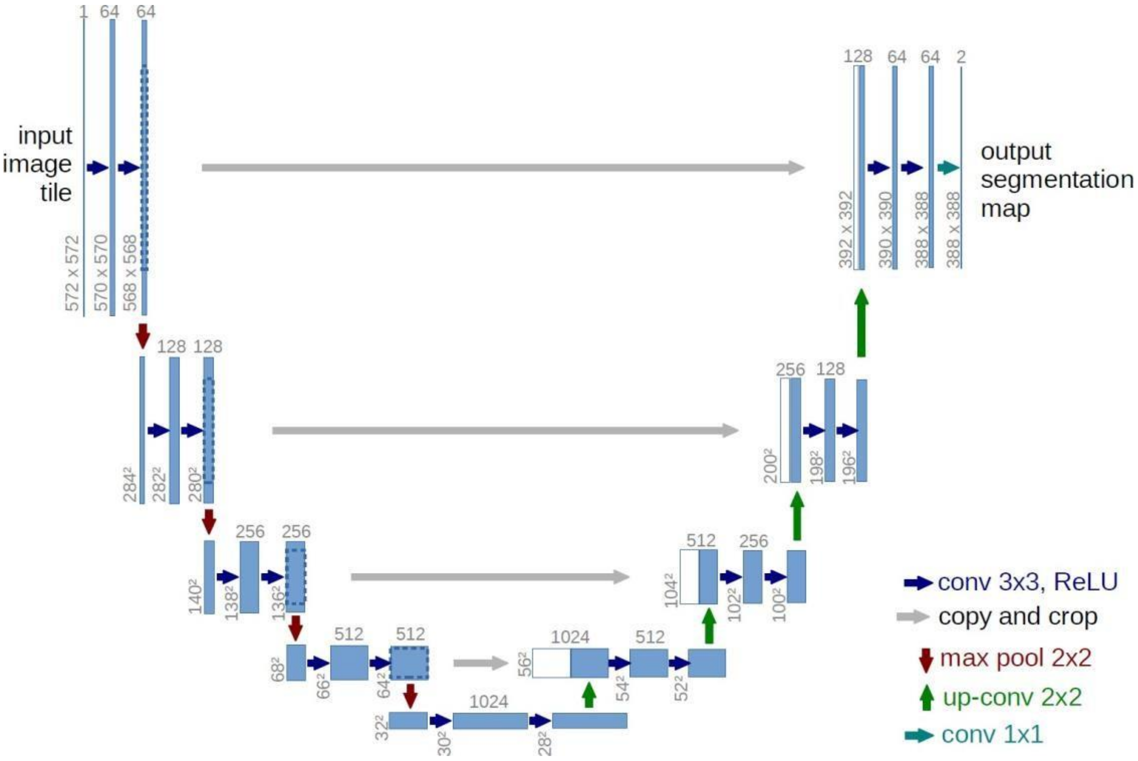

Figure 1: U-Net Network Architecture [14]

Advanced Models

Recent advances have seen improvements by fusing deep learning with other techniques, incorporating attention mechanisms and usage of transformer architectures.

- Attention U-Net: This model is a modification of the U-Net architecture which incorporates the attention gate mechanism to make the model focus on specific regions of the image thus improving the quality of the lesion segmentation [15]. The visualization of this network can be seen in Figure 2.

- Medical Transformer: Drawing inspiration from the success of transformers in NLP, transformers have been adapted to medical image analysis with superior lesion boundaries in the segmentations [16].

Several of these works will be compared to the final model which will be proposed for lung lesion segmentation. The attention mechanism is an important part of the networks which will be used, and the gates marked by AG can be seen in Figure 2 are used to allow the network to focus selectively on specific regions of the input data.

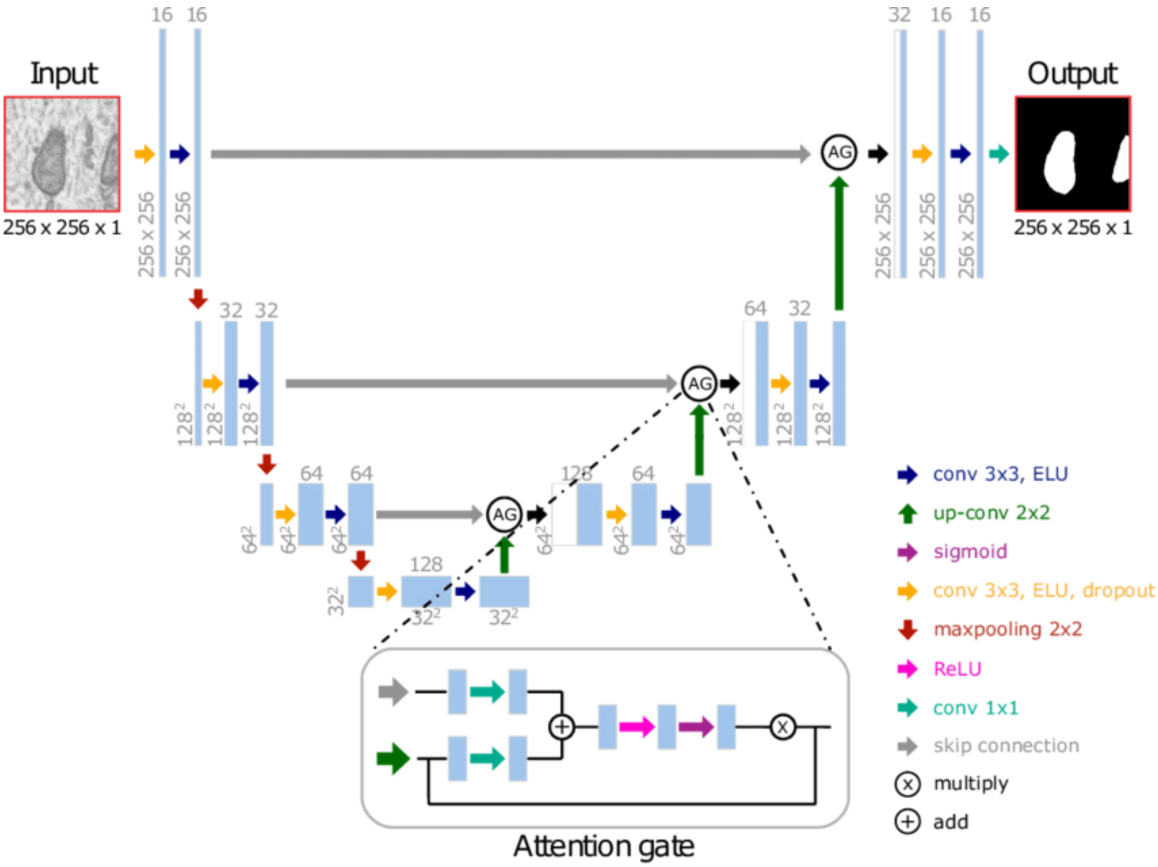

Figure 2. Attention U-Net Network Architecture [15]

CHAPTER THREE

DATA PREPARATION

Dataset

The LUNA16 dataset [17] used in this project was created in response to the challenges of early detection of lung lesions. While the Low-Dose Computed Tomography (LCDT) screen has been shown to reduce mortality from lung cancer, detecting the lung nodules themselves is challenging. Automatic nodule detection algorithms can be useful in improving the efficiency and accuracy of this process. The data in LUNA16 is sourced from the Lung Image Database Consortium (LIDC) - Image Database Resource Initiative (IDRI) dataset [6]. The LIDC-IDRI dataset consists of 1018 CT scan volumes, with annotations. Nodules are categorized based on their size:

- Small nodules: Less than 3mm (about 0.12 in)
- Medium nodules: Between 3mm (about 0.12 in) and 30mm (about 1.18 in)
- Large nodules: Greater than 30mm (about 1.18 in)

The focus of LUNA16 is primarily on the medium-sized nodules, as these are clinically significant and can be challenging to detect. The LIDC - IDRI annotation process involved four experienced thoracic radiologists. Each scan was independently reviewed by at least two radiologists. Disagreements between radiologists were resolved through an iterative consensus read. For LUNA16, nodules that were annotated by at least three out of four radiologists were considered positive examples, ensuring high confidence in the nodule labels. A few samples are visualized in Figure 3.

The LUNA16 dataset is a subset of the original LIDC - IDRI dataset, which contains 1,018 scans following pre-requisites resulting in 888 scans. There are a total of 1186 nodules in this dataset. Resolution varies between scans, but typically the in-plane resolution (x-y plane) is close to 0.7mm x 0.7mm. Regarding the slice thickness, it ranges from 0.6mm to 2.5mm, with most scans having a slice thickness between 1mm and 2mm. Finally, the scans themselves are typically 512 x 512 in the x-y plane, but the number of slices (z-dimension) varies based on the extent of the lungs and the slice thickness. For LUNA16, only nodules with a diameter between 3mm and 30mm are included, as these are the most clinically relevant.

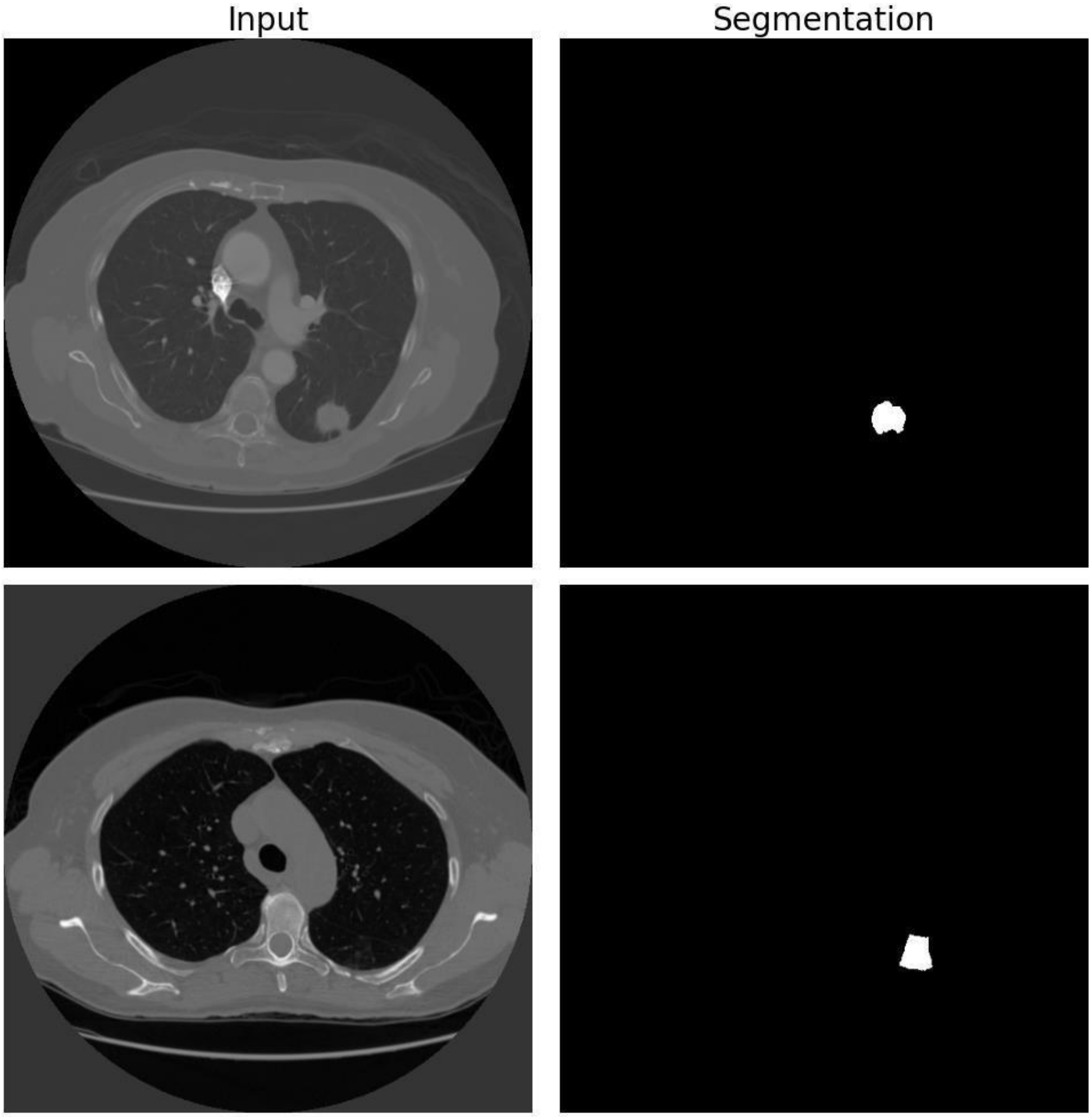

Figure 3. LUNA16 Lung Dataset Sample with Annotations [17]

Data Format

The dataset is available for download both on Kaggle [21] and the official LUNA16 website [22], and is provided in the Meta Image format, denoted by the

.mhd file extension. This format is commonly used for medical imaging datasets, as it supports a wide range of pixel data types and multi-dimensional images. The .mhd files serve as header files, describing the size, dimension, and data type of the image data, among other metadata. Accompanying each. mhd header file is a .raw file, which contains the actual image data in binary format. To process and read these files, the SimpleITK library is recommended. SimpleITK provides a straightforward interface for image I/O, and it abstracts away many of the complexities associated with medical image processing. Researchers and practitioners can easily load the LUNA16 scans into Python using SimpleITK, facilitating further analysis, visualization, and experimentation with the dataset.

Data Preprocessing

The data set was processed to extract relevant details for deep learning segmentation using the public library. Here are the following steps taken:

Nodule Malignancy Calculation

For each nodule, malignancy was calculated based on annotations made by radiologists. The median high value of the malignancy annotations was taken as the representative score. If this score was above 3, the nodule was labelled as cancerous; if below 3, it was labelled as non- cancerous. A score of exactly 3 was tagged as ambiguous.

Patients with Annotated Nodules

For patients with annotated nodules, the following steps were taken:

- The consensus function from pylidc was employed to generate a consensus mask, considering the given confidence level and padding.
- The mask was then used to extract the region of interest from the original DICOM volume.
- Each slice of the segmented nodule was processed individually. Slices with a mask size below the defined threshold were disregarded.
- For the remaining slices, the lung region was isolated using the segment_lung function.
- Both the segmented lung slice and its corresponding mask were saved as numpy arrays, with systematic naming conventions to ensure traceability.

Patients without Annotated Nodules

For patients without any annotated nodules:

- The dataset was considered "clean", and each slice was processed without the need for a consensus mask.
- The first 50 slices of the DICOM volume were segmented to retain only the lung region.
- A placeholder mask with no nodules (all zeros) was created for each slice.
- Both the segmented lung slice and its placeholder mask were saved with a naming convention denoting their "clean" status.

The lung data is visualized after said preprocessing in Figure 4.

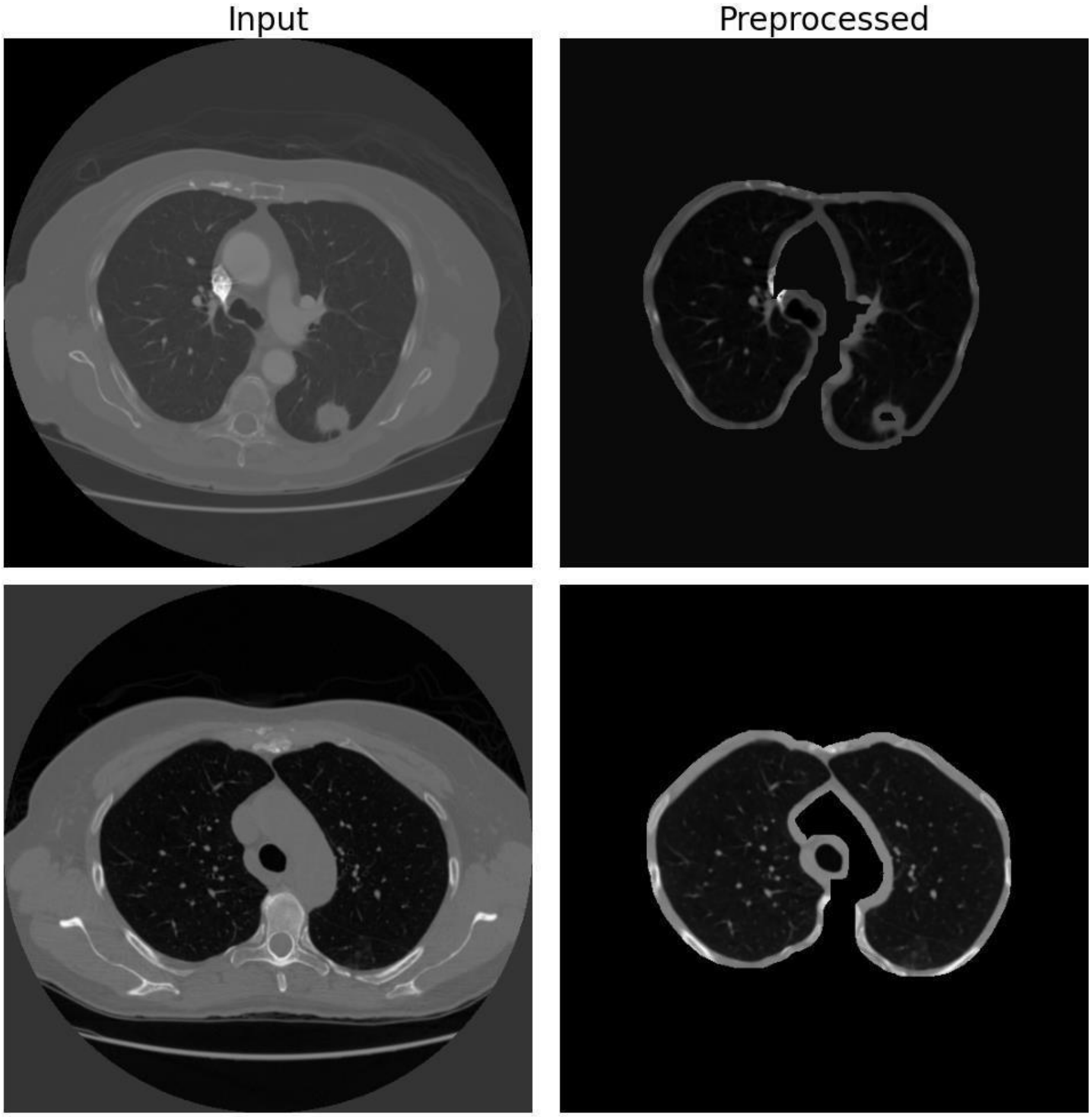

Figure 4. Input data visualized after preprocessing with pylidc

CHAPTER FOUR

METHODS

This chapter will outline the network architectures implemented for the task and their optimized hyperparameters. Further research improvements to the transformer-based architecture are also discussed.

Baseline U-Net Model

The U-Net architecture [14] as mentioned in Chapter 2 was originally proposed for biomedical image segmentation. This forms the baseline for the lung lesion segmentation task. The architecture can be briefly described as follows as can be seen in Figure 1.

U-Net consists of an encoding or downsampling path and a corresponding symmetric decoding or upsampling path connected by a bottom layer. Each layer in the encoding path consists of two consecutive 3x3 convolutions followed by a Rectified Linear Unit (ReLU) activation and a 2x2 max pooling operation with stride 2 for downsampling. The number of channels is doubled after each downsampling step.

The decoding path is like the encoding path. It comprises an upsampling of the feature map followed by a 2x2 convolution ("up-convolution"), a concatenation with the corresponding feature map from the encoding path, and two 3x3 convolutions each followed by a ReLU activation. The number of channels is halved after each upsampling step following the inverse of the encoder pipeline.

The entire architecture follows the visualization shown in Figure 1. This network will be used as a baseline to compare the improved performance and other considerations.

Hyperparameters

After several repeated experiments the optimal hyperparameters are chosen for good generalization. They are as follows:

Architecture Parameters

- Convolutional Layers: Two consecutive 3x3 convolutions per layer.
- Activation Function: ReLU
- Pooling Operation: 2x2 max pooling with a stride of 2 in the encoding path.
- Upsampling Operation: 2x2 up-convolution in the decoding path.
- Number of Channels: Doubled after each downsampling step and halved after each upsampling step.
- Loss Function: Pixel-wise softmax over the final feature map combined with the cross-entropy loss.
- Optimizer: Adam.
- Learning Rate: 1 × 10−4
- Batch Size: 32.
- Stopping Criterion: Convergence based on validation loss.

Attention U-Net for Lung Lesion Segmentation

The method used in this project uses the attention mechanism. The core principle behind this mechanism is to weigh the importance of different features in given data thereby allowing the model to focus on different regions of interest. In the context of medical image segmentation, this means emphasizing areas that are more likely to contain the target structures while suppressing the less relevant areas. The AG in the Attention U-Net computes attention coefficients, which are used to weigh the feature maps. The AG considers both the encoder's feature map and the corresponding feature map from the decoder path. Mathematically, given a feature map g from the decoder and a feature map x from the encoder, the attention coefficients α can be calculated as:

$$\alpha = \text{softmax}(Wgg + Wxx)$$

Where Wg and Wx are trainable weight matrices.

The attended feature map is then computed by element-wise multiplication of α with x:

$$x' = \alpha \odot x$$

In this network, the AG is introduced at every skip connection. As the U-Net decoder upsamples its feature maps, each upsampled map is merged with the corresponding encoder feature map after passing through an attention gate.

This let the network weigh the importance of features from the encoder path before merging, refining the feature fusion process and making it more adaptive to the specific input.

Hyperparameters

The Attention U-Net model's performance was optimized over several runs and the following hyperparameters were the best:

Table 1. Selected Hyperparameters for Attention U-Net Training.

Hyperparameter	Value
Depth of U-Net	4
Number of Initial Filters	64
Kernel Size	3 × 3
Activation Function	ReLU
Batch Size	32
Learning Rate	1 × 10−4
Loss Function	Cross Entropy Loss
Optimizer	Adam
Attention Gate Activation Function	Sigmoid
Dropout Rate	0.2
Weight Decay (L2 Regularization)	0.0005
Attention Type	Additive Attention
Up sampling Method	Transposed Convolution

Cross-Validation

In comparing the results of the various experiments, a 5 5-fold cross validation strategy was used to improve the reliability and accuracy of these metrics for the task of lung lesion segmentation. This is illustrated in Figure 5. Cross-validation is a widely used technique in machine learning for assessing how a model generalizes to independent datasets. More specifically, 5-fold cross-validation involves splitting the dataset into 5 equal-sized 'folds'. The training and evaluation process on train and validation sets is conducted 5 times with each fold acting as the validation set exactly once, and the remaining four folds collectively used as the training set.

The illustration in Figure 5 delineates the iterative process of this approach:

- In the first iteration, Fold 1 acts as the validation set while Fold 2, Fold 3, Fold 4, and Fold 5 together form the training set.
- In the subsequent iteration, Fold 2 is utilized as the validation set and the other folds constitute the training data.
- This pattern continues iteratively until each fold has been used as the validation data exactly once.

Finally, after the relevant model parameters are chosen the performance is evaluated on the test set as the result of that model.

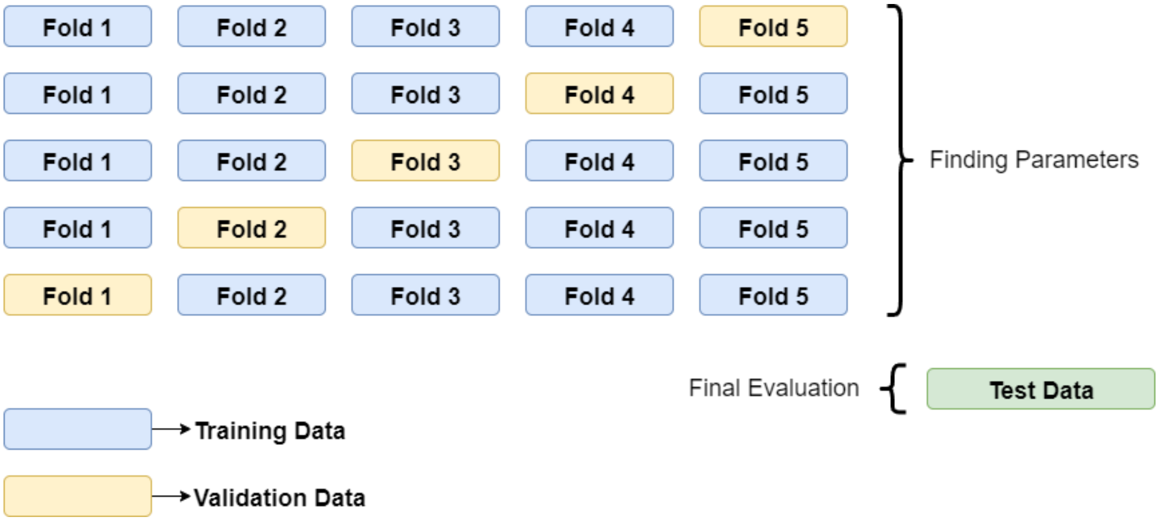
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Finding Parameters
Final Evaluation
Test Data
Training Data
Validation Data

Figure 5. 5-fold Cross Validation Illustration

CHAPTER FIVE

RESULTS AND ANALYSIS

Metrics

This section will outline the metrics used to compare the methods implemented for the lung lesion segmentation task.

Dice Score

The Dice Score is a vital metric for evaluating segmentation models in medical imaging. It's representative of how well the model is segmenting the required anatomy compared to the ground truth.

Mathematically, the Dice Score D can be defined as follows:

$$D = \frac{2 \times |A \cap B|}{|A| + |B|}$$

Where A is the set of pixels in the predicted segmentation and B is the set of pixels in the ground truth. The Dice Score ranges between 0 and 1 where a score of 1 represents a perfect segmentation and a score of 0 indicates no overlap.

Jaccard Index / Intersection Over Union (IoU)

Jaccard Index, often synonymous with IoU is another useful metric for evaluating the quality of segmentations by the different models in medical imaging. This metric is like the Dice Score and quantifies the overlap between the predicted segmentation and ground truth.

Mathematically, the Jaccard Index is defined as:

$$J = \frac{|A \cap B|}{|A \cup B|}$$

Where A and B represent the sets of pixels in the predicted and ground-truth segmentations, respectively. The Jaccard Index also ranges from 0 to 1, with a score of 1 indicating perfect overlap and a score of 0 suggesting no overlap.

Results

The methods outlined in Chapter 4 were employed on the LUNA16 dataset described earlier for lung lesion segmentation. This chapter will give a brief overview of the different results obtained by the implemented models concerning the two different metrics. A comprehensive review of the performance of the deep learning models will be presented. The primary goal is to check how the final model is compared to the baseline U-Net model.

Before discussing in-depth results, it is essential to note the number of trainable parameters in the different models:

Table 2. Comparison of Trainable Parameters for Baseline UNet and Attention U-Net.

Model	Number of trainable parameters
Baseline U-Net	19,017,556
Attention U-Net	22,133,455

Table 2 shows that the number of trainable parameters for the Attention U-Net is close to the baseline model with the added attention gates at the skip connections as shown in Figure 2. The performance results with the described metrics are shown in Table 3. The optimized models with corresponding hyperparameters for both the baseline and the attention U-Net show that the dice score and IoU are both improved with the introduction of attention gates. Both the dice score and IoU being positively linked show improved performance in comparison.

Table 3: Experimental Results of Segmentation Models

Model	Dice Score	IoU
Baseline U-Net	0.885 ± 0.036	0.873 ± 0.041
Attention U-Net	0.909 ± 0.029	0.899 ± 0.033

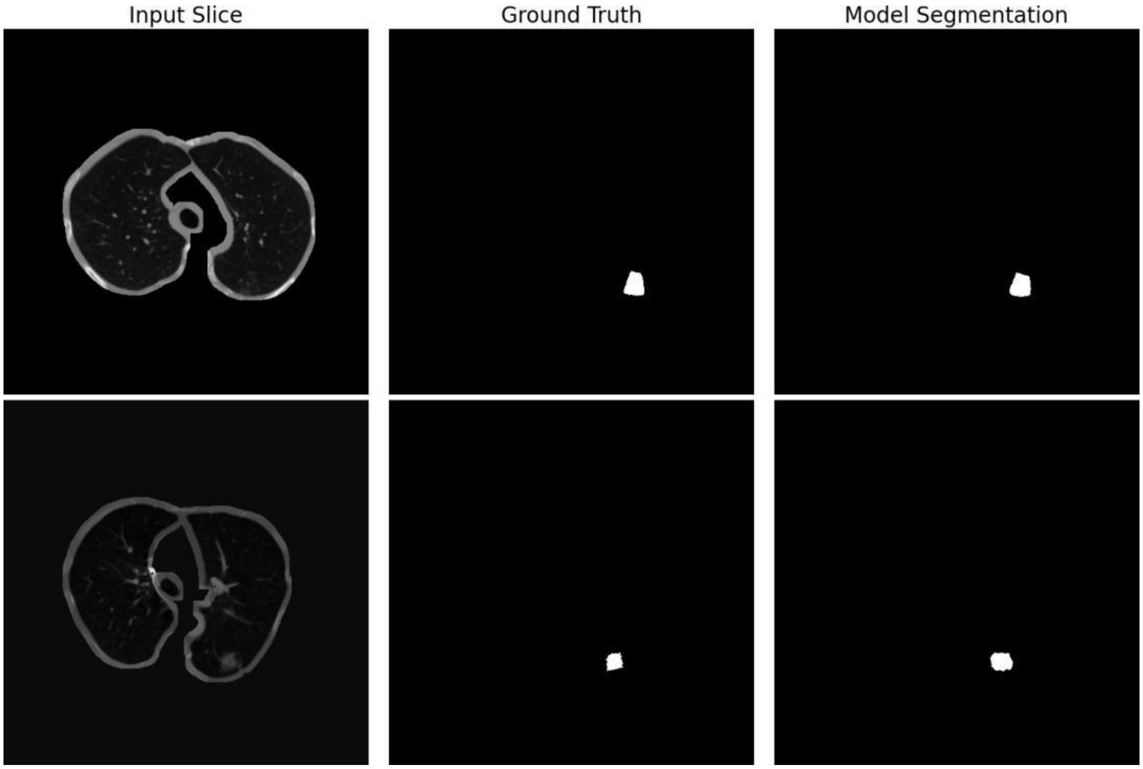

Figure 6. Two Exemplary Cases of Segmentation Output

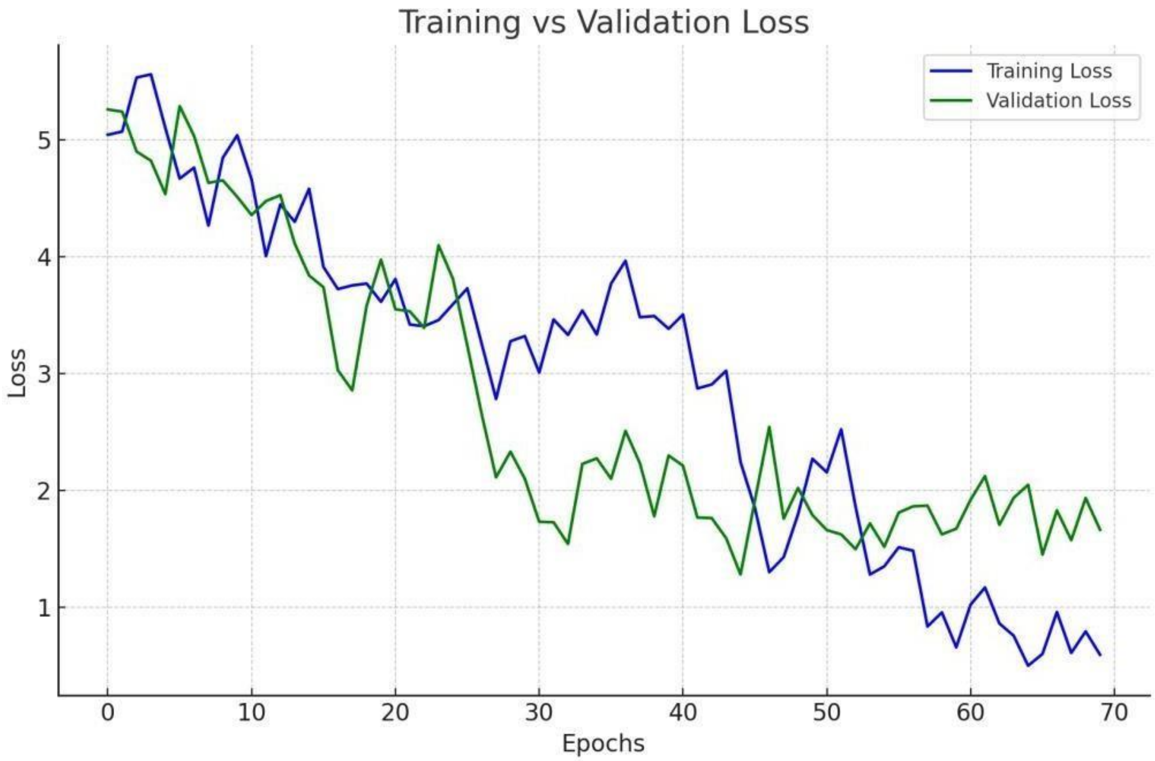

Figure 7. Training and Validation Curves of the Attention U-Net Model over 70 Epochs

Table 4. Training vs Validation Loss

Epoch	Training Loss	Validation Loss
1	5.08693	4.99226
2	5.6226	5.59911
3	5.61803	5.34333
4	5.23742	5.08184
5	5.41091	4.72258
6	5.0313	4.29719
7	4.2583	4.77496
8	4.00795	4.73594
9	3.55563	4.25451
10	3.67653	3.95932
11	3.85388	4.04657
12	4.45717	4.6387
13	4.45636	4.30324
14	4.27372	3.7095
15	4.44216	3.73691
16	3.62965	3.68958
17	3.77666	2.88572
18	4.02188	3.22363
19	3.50043	3.28665
20	3.90386	3.54337
21	3.92643	3.9646
22	3.69783	3.90557
23	3.83397	3.40447
24	4.40237	2.84969
25	4.15535	3.62332
26	3.98408	3.20296
27	3.63366	2.92625
28	3.25296	2.72353
29	3.47226	2.6359
30	3.10158	2.43641
31	3.10214	2.15493
32	2.91951	1.96929
33	3.06039	1.97776
34	3.09716	1.77585
35	3.31579	1.57113
36	2.98631	1.67725
37	2.80946	1.67588

38	2.45473	2.28223
39	2.3485	1.30761
40	2.50236	1.87955
41	2.7057	2.31814
42	2.32604	1.38801
43	2.57954	1.57943
44	2.88455	1.56939
45	2.79713	1.31012
46	3.15668	1.39038
47	2.59493	1.26781
48	2.04218	2.11187
49	1.47678	2.34657
50	2.04517	2.45075
51	2.17228	2.44808
52	2.03001	1.75795
53	2.01282	1.72635
54	1.56089	2.01882
55	2.29765	1.43315
56	2.10017	1.90242
57	1.93772	1.87789
58	1.99065	1.75192
59	1.93602	2.06288
60	1.80238	2.07524
61	1.38626	1.89876
62	1.4553	2.23723
63	1.90854	1.80658
64	2.04656	2.0216
65	2.19924	2.14925
66	1.65654	1.90273
67	1.11487	1.98512
68	0.97659	1.55779
69	0.91598	2.03664
70	1.16683	1.88194

Figure 6 shows two exemplary cases of lung lesion segmentations next to the ground truth. As can be seen, while the general position of the lesion is well segmented, there is an overall difference in the boundary of the segmentation which contributes to the dice score shown in Table 3. Following this, the training and validation curves for the attention U-Net model can be seen visualized in Figure 7 and shown explicitly in the following table.

Insights And Project Contributions

This section will cover the insights provided by the models trained in this project along with what the novelties are in the network architecture compared to the original models that are used. Firstly, the depth of the UNet was modified to four levels, which is different from the original architecture that often varied in depth. This project specific tuning is related to the complexity of lung images, suggesting that a four-level depth is sufficient for capturing the necessary hierarchical features of lung lesions without overfitting, given the size and variation within the LUNA16 dataset. The number of initial filters was set at 64, which is a significant difference from the UNet's original configuration that might start with a smaller number of filters. This increase addresses the need for a broader initial feature extraction capability, which is likely useful for identifying the complex features of the lung lesions against different densities that are found in CT scans.

Along with this, using a dropout rate of 0.2 and L2 regularization with a weight decay of 0.0005 in the Attention UNet model are additions to the original paper. These regularization techniques are particularly relevant for medical imaging tasks where the training datasets are often limited in size, increasing the risk of overfitting. The resulting architecture makes the model more able to generalize from the training data to data that is not seen which is important for clinical applications. Moreover, a transposed convolution is used for upsampling in the Attention UNet with a preference for more complex and learnable upsampling operations over simpler bilinear upsampling methods that were originally used. This choice is made mainly to improve the precision of upsampled feature maps which can result in more accurate lesion boundaries which is the main objective of this segmentation task. These hyperparameter choices reflect an adaptation of the UNet and Attention UNet models for lung lesion segmentation, considering the dataset's specific challenges.

Comparison With Other Published Results

While several results have been published with lung segmentation dice scores being above 98%, lesion segmentation accuracy is the most challenging task. The following are some published results on the same dataset.

Table 5. Comparison with Results Published on the same Dataset

Model	Dice Score
Keetha, Annavarapu, et al., 2020	0.8282
Banu et al., 2021	0.8979
Sathish et al., 2020	0.9102
Proposed Model	0.9091

As can be seen in Table 5, the segmentation results of the given attention U- Net come close to if not better than published results on the same dataset. There can be further experiments conducted on different datasets to validate these results further in future work.

CHAPTER SIX

CONCLUSION

This report presented a deep learning-based approach for lung nodule segmentation. Performance metrics, including Dice Score, and IoU validate the model's effectiveness when compared to the baseline U-Net model. Several runs of the model were trained before selecting the final hyperparameters of the Attention U-Net model. It was also observed that for relatively smaller nodules, the performance is improved significantly compared to the baseline which is an improvement in overall performance.

While the model demonstrated strong performance in most of the cases, the challenging delineations to automate are lesions which are not spherical, lie at the border of other classes of the scan and are smaller in size. These tend to contribute to false positives or negatives bringing down the overall dice. Regardless, the optimized Attention U-Net model was able to achieve an overall dice score of 90.9% which is valuable for automating this task to be used in conjunction with the radiologist's opinions.

Another potential enhancement is the utilization of advanced attention mechanisms. Given the complex nature of medical scans, models that effectively capture relationships between different parts of the volume with multi-head attention could be beneficial.

In summary, this research emphasized the utility of a finetuned Attention U-Net model for lung nodule segmentation. By building on these methodologies and refining the approach, the segmentation process can be optimized, leading to reduced manual intervention and increased accuracy in delineation.

APPENDIX A

DATASET DESCRIPTION

This Appendix provides details on the dataset used in this project. The LUNA16 dataset contains 888 scans of patient data which contain 1186 nodules for the entire dataset. They are available in MetaImage format in .mhd files and with the corresponding annotated segmentation. Resolution in-plane is close to mm x 0.7mm and slice thickness ranges from 0.6mm to 2.5mm

APPENDIX B

MODEL ARCHITECTURE DETAILS

The final model used in this study is a finetuned attention U-Net architecture using attention gates at skip connections in a U-Net. The following are the final hyperparameters of the optimized attention U-Net model:

- Convolutional Layers: Two consecutive 3x3 convolutions per layer.
- Activation Function: ReLU
- Pooling Operation: 2x2 max pooling with a stride of 2 in the encoding path.
- Upsampling Operation: 2x2 up-convolution in the decoding path.
- Number of Channels: Doubled after each downsampling step and halved after each upsampling step.

APPENDIX C

TRAINING PARAMETERS

The following training hyperparameters were used for the final model that was optimized:

- Optimizer: Adam
- Learning Rate
- Batch Size: 32
- Stopping Criterion: Convergence based on validation loss

APPENDIX D

MODEL TRAINING AND EVALUATION CODE

The checkpoints for the various trained models can be provided upon request. This includes network code, checkpoints, training, and evaluation files. The final proposed model code can be seen below:

```
import numpy as np
import SimpleITK as sitk
import pylidc as pl
import torch
import torch.nn as nn
from sklearn.model_selection import KFold
from torch.utils.data import DataLoader, TensorDataset

# Load LUNA16 .mhd files
def load_itk_image(filename):
     itkimage = sitk.ReadImage(filename)
     numpy_image = sitk.GetArrayFromImage(itkimage)
     numpy_origin =
np.array(list(reversed(itkimage.GetOrigin())))
     numpy_spacing =
np.array(list(reversed(itkimage.GetSpacing())))
     return numpy_image, numpy_origin, numpy_spacing

def get_segmented_lungs(scan, plot=False):
    # This preprocessing is done according to the
specifications mentioned before

# Load dataset
data_path = 'path_to_LUNA16_data'
filenames = [...]
images = []
masks = []

for file in filenames:
    img, origin, spacing = load_itk_image(file)
    segmented_lungs = get_segmented_lungs(img)
    images.append(img)
    masks.append(segmented_lungs)

class ConvBlock(nn.Module):
    def init (self, in_channels, out_channels):
        super(ConvBlock, self). init ()
        self.block = nn.Sequential(
            nn.Conv2d(in_channels, out_channels,
        kernel_size=3, padding=1),
            nn.ReLU(),
               nn.Conv2d(out_channels, out_channels,
          kernel_size=3, padding=1),
               nn.ReLU()
)
     def forward(self, x):
          return self.block(x)


class AttentionUNet(nn.Module):
     def init (self, in_channels, out_channels):
          super(AttentionUNet, self). init ()

          # Encoder
          Self.enc1 = ConvBlock(in_channels, 64)
          self.enc2 = ConvBlock(64, 128)
          self.enc3 = ConvBlock(128, 256)
          self.enc4 = ConvBlock(256, 512)

          self.pool = nn.MaxPool2d(kernel_size=2, stride=2)

          # Decoder
          self.up3 = nn.ConvTranspose2d(512, 256,
     kernel_size=2, stride=2)
          self.dec3 = ConvBlock(512, 256)
```

```
        self.attn3 = AttentionBlock(256, 256, 128)
        self.up2 = nn.ConvTranspose2d(256, 128,
kernel_size=2, stride=2)
        self.dec2 = ConvBlock(256, 128)
        self.attn2 = AttentionBlock(128, 128, 64)

        self.up1 = nn.ConvTranspose2d(128, 64,
    kernel_size=2, stride=2)
        self.dec1 = ConvBlock(128, 64)
        self.out = nn.Conv2d(64, out_channels,
    kernel_size=1)

    def forward(self, x):
        # Encoder
        x1 = self.enc1(x)
        x2 = self.enc2(self.pool(x1))
        x3 = self.enc3(self.pool(x2))
        x4 = self.enc4(self.pool(x3))

        # Decoder with attention
        x_up = self.up3(x4)
        x3 = self.attn3(x_up, x3)
```

```
          x3 = self.dec3(torch.cat([x_up, x3], dim=1))

          x_up = self.up2(x3)
          x2 = self.attn2(x_up, x2)
          x2 = self.dec2(torch.cat([x_up, x2], dim=1))

          x_up = self.up1(x2)
          x1 = self.dec1(torch.cat([x_up, x1], dim=1))
          return self.out(x1)

images = np.array(images)[:, np.newaxis, :, :]
masks = np.array(masks)[:, np.newaxis, :, :]

images_tensor = torch.tensor(images, dtype=torch.float32)
masks_tensor = torch.tensor(masks, dtype=torch.float32)

dataset = TensorDataset(images_tensor, masks_tensor)

num_epochs = 70

kf = KFold(n_splits=5, shuffle=True, random_state=42)

for fold, (train_idx, val_idx) in
```

```
enumerate(kf.split(images)):
     train_dataset =
TensorDataset(images_tensor[train_idx],
masks_tensor[train_idx])
     val_dataset = TensorDataset(images_tensor[val_idx],
masks_tensor[val_idx])

     train_dataloader = DataLoader(train_dataset,
batch_size=32, shuffle=True)
     val_dataloader = DataLoader(val_dataset,
batch_size=32, shuffle=False)
     for epoch in range(num_epochs):
          model.train()
          train_loss = 0
          for img, mask in train_dataloader:
               img, mask = img.to(device), mask.to(device)
          optimizer.zero_grad()
               outputs = model(img)
               loss = criterion(outputs, mask)
               loss.backward()
               optimizer.step()
               train_loss += loss.item()
          # Validation loop
          model.eval()
          val_loss = 0
          with torch.no_grad():
               for img, mask in val_dataloader:
                    img, mask = img.to(device),
               mask.to(device)
                    outputs = model(img)
                    loss = criterion(outputs, mask)
                    val_loss += loss.item()

          print(f"Fold {fold+1}, Epoch {epoch+1},
          Train Loss: {train_loss/len(train_dataloader)},
          Val Loss: {val_loss/len(val_dataloader)}")

          # Save model checkpoint
          torch.save(model.state_dict(),
f"attention_unet_fold{fold+1}_epoch{epoch+1}.pth")
```

REFERENCES

[1] Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6), 641-647.

[2] Armato, S. G., Giger, M. L., Moran, C. J., Blackburn, J. T., Doi, K., & MacMahon, H. (1999). Computerized Detection of Pulmonary Nodules on CT Scans. *Radiographics*, 19(5), 1303-1311.

[3] Atharv C. (2020) Kaggle Data Download URL: https://www.kaggle.com/datasets/avc0706/luna16

[4] Banu, S. F., Sarker, M. M. K., Abdel-Nasser, M., Puig, D., & Raswan, H. A. (2021). Aweu-net: An Attention-Aware Weight Excitation U-Net for Lung Nodule Segmentation. *Applied Sciences*, 11(21), 10132.

[5] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). 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.

[6] Chen, C., Qin, C., Qiu, H., Tarroni, G., Duan, J., Bai, W., & Rueckert, D. (2020). Deep Learning for Cardiac Image Segmentation: *A Review. Frontiers in Cardiovascular Medicine*, 7, 25.

[7] Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20, 273-297.

[8] Gonzalez, Rafael C and Richard E Woods (2002). “Digital Image Processing. Upper Saddle River”. In: *J.: Prentice Hall.*

[9] Gould, M. K., Donington, J., Lynch, W. R., Mazzone, P. J., Midthun, D. E., Naidich, D. P., & Wiener, R. S. (2013). Evaluation of Individuals with Pulmonary Nodules: When is it Lung Cancer?: Diagnosis and Management of Lung Cancer: American College of Chest Physicians Evidence-Based Clinical Practice Guidelines. *Chest*, 143(5), e93S-e120S.

[10] Haralick, R. M., & Shapiro, L. G. (1985). Image Segmentation Techniques. *Computer Vision, Graphics, and Image Processing*, 29(1), 100-132.

[11] Haralick, R. M. (1979). Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5), 786-804.

[12] Keetha, N. V., & Annavarapu, C. S. R. (2020). U-Det: A Modified U-Net Architecture with Bidirectional Feature Network for Lung Nodule Segmentation. *arXiv preprint arXiv*:2003.09293.

[13] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A Survey on Deep Learning in Medical Image Analysis. *Medical Image Analysis*, 42, 60-88.

[14] National Lung Screening Trial Research Team. (2011). Reduced Lung-Cancer Mortality with Low-Dose Computed Tomographic Screening*. New England Journal of Medicine*, 365(5), 395-409.

[15] Revel, M. P., Bissery, A., Bienvenu, M., Aycard, L., Lefort, C., & Frija, G. (2004). Are Two-Dimensional CT Measurements of Small Noncalcified Pulmonary Nodules Reliable?. *Radiology*, 231(2), 453-458.

[16] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). *Springer International Publishing.*

[17] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention U-Net: Learning where to Look for the Pancreas. *arXiv preprint arXiv*:1804.03999.

[18] Sahiner, B., Hadjiiski, L. M., Chan, H. P., Shi, J., Way, T., Cascade, P. N., ... & Wei, J. (2007, March). The Effect of Nodule Segmentation on the Accuracy of Computerized Lung Nodule Detection on CT Scans: Comparison on a Data Set Annotated by Multiple Radiologists. In Medical Imaging 2007: Computer-Aided Diagnosis (Vol. 6514, pp. 175-181). *SPIE*.

[19] Samuel, G. (2011). The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans. *Medical Physics*, 38, 2.

[20] Sathish, R., Sathish, R., Sethuraman, R., & Sheet, D. (2020, July). Lung Segmentation and Nodule Detection in Computed Tomography Scan Using a Convolutional Neural Network Trained Adversarially using Turing

Test Loss. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (Pp. 1331-1334). *IEEE*.

[21] Setio Et Al. In Validation, Comparison, and Combination of Algorithms for Automatic Detection of Pulmonary Nodules in Computed Tomography Images: The Luna16 Challenge URL: https://luna16.grand-challenge.org/Data/

[22] Setio, A. A. A., Traverso, A., De Bel, T., Berens, M. S., Van Den Bogaard, C., Cerello, P., ... & Jacobs, C. (2017). Validation, Comparison, and Combination of Algorithms for Automatic Detection of Pulmonary Nodules in Computed Tomography Images: The LUNA16 Challenge. *Medical Image Analysis*, 42, 1-13.