

# Applications of Synthetic Data Integration for Deep Learning for Volumetric Analysis and Segmentation in Thoracic CT Imaging

Ali Zeyrek<sup>1</sup> and Sergio M Navarro, MD MBA<sup>2,3</sup>

<sup>1</sup>Cicekli AI, Cicekli, Izmir, TR

<sup>2</sup>University of Minnesota, Department of Surgery, Minneapolis, MN, US

<sup>3</sup>Mayo Clinic, Department of Surgery, Rochester, MN, US

\*Corresponding author: [navar176@umn.edu](mailto:navar176@umn.edu)

## Abstract

This study presents a framework for processing Digital Imaging and Communications in Medicine (DICOM) medical imaging data by integrating synthetic objects for volumetric analysis and simulation for applications in assessment of computed tomography (CT) imaging used in thoracic surgery. Functions are designed to generate synthetic objects including geometric shapes such as spheres, cubes, rectangular prisms, cylinders, and blobs with known volumes. Validation is performed through test functions to ensure accuracy and consistency. Additionally, the use of UNet models for segmenting various chest pathologies, such as hemothorax and pneumothorax, as well as organs, is demonstrated. The created framework is used to generate synthetic data to address the scarcity of publicly available hemothorax CT imaging data. Models achieved high performance, assessed by various metrics. The framework and models provide a robust tool for data augmentation and analysis in medical imaging, potentially enhancing clinical decision-making and supporting research in thoracic surgery and related fields.

**Keywords:** computed tomography; DICOM; volumetric assessment; thoracic surgery; Monte Carlo method; data augmentation; deep learning; segmentation.

## 1. Introduction

Understanding the volume of hemothorax and other various lesions, and measuring their volumes is crucial for determining a diagnosis. This paper details a general purpose framework for synthetic data generation that processes DICOM medical imaging data, creates synthetic objects such as spheres and blobs, and estimates their volumes using the Monte Carlo method. Using this framework, this paper illustrates and explains the training of segmentation models for chest organs, hemothorax and pneumothorax.

## 2. Methods

### 2.1 Reporting Guidelines

The study followed appropriate research reporting guidelines, including relevant EQUATOR research reporting checklists and other pertinent reporting as applicable. Data sources are referenced and only publicly available data sources were used for the study, with Institutional Research Board exemption granted based on checklist review. This work used ONLY simulated data or ONLY data that was openly available to the public before the initiation of the study.

**NOTE: This preprint reports new research that has not been certified by peer review and should not be used to guide clinical practice.**

## 2.2 Synthetic Hemothorax Data Framework 35

Synthetic hemothorax data framework enables us to overcome the scarcity of publicly available hemothorax data. Using the tool we created, we can synthetically place specific 3D objects in DICOM series, on any arbitrary coordinates and dimensions, with flexible options for setting object boundaries and Hounsfield values. 36  
37  
38  
39

## 2.3 Data Loading 40

Patient data is loaded and organized using the `load_patient_data` function, which retrieves DICOM series from a specified folder. Function preserves metadata and reads multiple series if there are for a patient. 41  
42

## 2.4 Object Declaration 43

In the class `Object`, there are functions for multiple objects: sphere, quadrilateral, cylinder, blob. Besides the required arguments for all, such as the position  $(z,y,x)$ , HU baseline value, HU noise amplitude and object noise, shapes may also require distinct arguments. The purpose of the functions are essentially to check whether a certain point is within the bounds of the object. 44  
45  
46  
47

### 2.4.1 Sphere 48

The `Sphere` function creates a sphere function based on the given radius and center coordinates. Function checks whether the distance  $r$  from a point  $(x, y, z)$  to the center  $(x_0, y_0, z_0)$  is within a given radius  $R$ , with added random noise from 'obj\_noise'. This effectively defines a noisy spherical boundary around the center point. 49  
50  
51  
52

### 2.4.2 Prism 53

In addition to regular arguments, the `Prism` function requires height, length and width. Function checks whether the coordinates  $(x, y, z)$  fall within the 3D rectangular region centered at  $(x_0, y_0, z_0)$ , with dimensions defined by 'width', 'length', and 'height'. The 'obj\_noise' term introduces random variations to each boundary, also simulating noise or irregularity if requested. 54  
55  
56  
57

### 2.4.3 Cylinder 58

The `Cylinder` function requires height and radius besides regular inputs. Function checks if a point  $(x, y, z)$  lies within a noisy cylindrical region centered at  $(x_0, y_0, z_0)$ , with a specified radius  $R$  and height  $h$ . It considers both the radial distance from the center in the  $xy$ -plane and the vertical position in the  $z$ -axis, adding specified noise to simulate irregular boundaries. 59  
60  
61  
62

### 2.4.4 Blob 63

`Blob` function evaluates whether a point  $(x, y, z)$  falls within a 3D Gaussian distribution centered at  $(x_0, y_0, z_0)$ , characterized by amplitudes  $A$ , standard deviations  $(\sigma_z, \sigma_y, \sigma_x)$ , and a threshold. It introduces noise to the threshold, creating a "blobby" region with randomized intensity variations. 64  
65  
66

## 2.5 DICOM Utilities 67

For making the tool capable, there are various utilities besides object types. User can define a HU value for the object, and optionally make it noisy by giving a variability amplitude for a normal function. Also, user can define object threshold noise, which makes the bounds of the object noisier. 68  
69  
70

## 2.5.1 Hounsfield Unit (HU) 71

### Definition 72

In computed tomography (CT), the **Hounsfield Unit (HU)** is used as a standardized measure of tissue 73  
density. It is derived from the linear attenuation coefficients of tissues, normalized such that the density of 74  
distilled water at standard pressure and temperature is set to **0 HU**, while the air density is set to **-1000** 75  
**HU**. This scale enables the differentiation of various body tissues based on their density. 76

The HU value is calculated from the linear attenuation coefficient ( $\mu$ ) using the formula: 77

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

Where: -  $\mu$  = attenuation coefficient of the tissue, -  $\mu_{\text{water}}$  = attenuation coefficient of water. 78

Thus, HU values quantitatively represent tissue densities in CT images, ranging from **-1000 HU** (air) 79  
to more than **1000 HU** (bone), with other tissues like fat, muscle, and blood having values between these 80  
extremes. 81

### Applying HU to Simulated Objects 82

When a simulated object is introduced into a CT volume, its density needs to be represented accurately 84  
in terms of Hounsfield Units to maintain the realism of the imaging. In this case, the simulated object's 85  
HU value is defined by a user-specified mean value, typically corresponding to the expected density of the 86  
simulated tissue (e.g., **soft tissue = 30 HU**, **bone = 700 HU**). To introduce realistic variation, Gaussian 87  
noise is added: 88

$$\text{HU}_{\text{noisy}} = \text{HU}_{\text{mean}} + N(0, \sigma)$$

Where: -  $\text{HU}_{\text{noisy}}$  = modified HU value with noise, -  $\text{HU}_{\text{mean}}$  = mean HU value set for the object, - 89  
 $N(0, \sigma)$  = Gaussian noise with mean 0 and standard deviation  $\sigma$ . 90

The addition of Gaussian noise models the natural variability seen in real tissue densities, ensuring that 91  
the object integrates more naturally into the CT image. 92

### Conversion of HU to Pixel Values 93

CT images typically store pixel values that are not directly in HU but require conversion using linear 95  
scaling parameters provided in DICOM metadata: **Rescale Slope** and **Rescale Intercept**. The conversion 96  
from HU to DICOM pixel values is described by: 97

$$\text{Pixel Value} = \frac{\text{HU} - \text{Rescale Intercept}}{\text{Rescale Slope}}$$

Where: - **Rescale Slope** and **Rescale Intercept** are constants provided in the DICOM header. 98

This conversion ensures that the modified object's density is accurately represented in the format used 99  
by CT imaging, making it possible to simulate the appearance of the object as it would be seen in a clinical 100  
scan. 101

## 2.6 Object Addition

First, z,y and x dimensions of the size of DICOM series is represented as an array from 0 to n. Then, each dimension is multiplied by the PixelSpacing or SliceThickness values to get actual coordinates. Using the coordinates, a meshgrid is created. Using the meshgrid, a 5D array is created for coupling voxel centers with their corner coordinates. Corner coordinates are used for ensuring the voxel is within the object bounds completely. The voxels with all corners within the object is added to the mask.

The created objects are added into the DICOM series for the patient within the thoracic space by creating a new DICOM database. In the new database, it is indicated that slices were altered.

For a processed series, before the addition of the hemothorax, the output of the chest model is obtained first. The process is explained in later sections. This is done to ensure that the object going to be added to stay within the bounds of the lung. To achieve this, hemothorax mask and lung mask for each slice is intersected. The output is then saved as the artificial hemothorax. This ensures that artificial hemothorax is placed within the lung space.

### 2.6.1 Volume Calculation

For calculating the volume of object, a minimum bounding box around the object is defined. Then, the volume is calculated with the following methodology.

The Monte Carlo method is used to estimate the volume of irregular objects, leveraging random sampling to determine the inclusion of points within the blob. The mathematical foundation of this approach is as follows:

Let  $f(x, y, z)$  be the blob function that defines the shape of the object, where:

$$f(x, y, z) = \begin{cases} 1 & \text{if } (x, y, z) \text{ is inside the object} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The volume  $V$  of the object can be expressed as a triple integral:

$$V = \int \int \int_{\Omega} f(x, y, z) dx dy dz \quad (2)$$

where  $\Omega$  is the bounding box containing the object.

The Monte Carlo method approximates this integral by randomly sampling points within the bounding box. If we generate  $N$  random points  $(x_i, y_i, z_i)$  uniformly distributed in  $\Omega$ , the volume can be estimated as:

$$V \approx V_{\Omega} \cdot \frac{1}{N} \sum_{i=1}^N f(x_i, y_i, z_i) \quad (3)$$

where  $V_{\Omega}$  is the volume of the bounding box.

In our implementation, we account for the pixel spacing and slice thickness of the DICOM images. The volume of a single voxel is calculated as:

$$V_{voxel} = \text{pixel\_spacing}_x \cdot \text{pixel\_spacing}_y \cdot \text{slice\_thickness} \quad (4)$$

The bounding box volume is then:

$$V_{\Omega} = (2 \cdot \max\_boundary_x) \cdot (2 \cdot \max\_boundary_y) \cdot (2 \cdot \max\_boundary_z) \cdot V_{voxel} \quad (5)$$

Finally, the estimated object volume is calculated as:

$$V_{object} = V_{\Omega} \cdot \frac{\text{inside\_count}}{N} \quad (6)$$

where `inside_count` is the number of sampled points that fall inside the object. 131

This Monte Carlo approach provides an efficient and flexible method for estimating the volume of irregular 132  
objects, with accuracy improving as the number of samples increases. 133

After the volume is calculated, it is multiplied by the ratio of pixels before and after to the lung and 134  
hemothorax mask intersection, so that volume by pixel information is accurate. 135

### 3. Deep Learning 136

Deep learning is used in multiple parts of this study. 137

#### 3.1 Definitions and Explanations of the Components and Metrics 138

##### 3.1.1 Dataset 139

##### CT Thorax Imaging 140

The original dataset used for thoracic CT imaging model development is available publicly on the Harvard 141  
Dataverse (Mostafavi, 2021). The dataset consists of CT thoracic imaging of more than 1,000 patients with 142  
a confirmed COVID-19 diagnosis. 143

##### 3.1.2 Models 144

##### UNET 145

The UNet architecture, first proposed by Ronneberger et al., 2015, is a fully convolutional network widely 146  
recognized for its efficiency in medical image segmentation. It features a symmetric "U-shaped" design, com- 147  
prising an encoder-decoder structure. The encoder progressively reduces spatial dimensions while extracting 148  
semantic features, utilizing convolutional and max-pooling layers. Conversely, the decoder employs trans- 149  
posed convolutions to restore spatial resolution, incorporating skip connections from the encoder to preserve 150  
high-resolution features. In this study, the UNet model was configured with the EfficientNet-B2 backbone, 151  
which is chosen for its balance between accuracy and computational efficiency. The model is optimized using 152  
pre-trained weights on the ImageNet dataset, enabling faster convergence and better performance in the 153  
segmentation of thoracic structures. 154

**ResNet** ResNet (Residual Network, He et al., 2015) ( is leveraged within the UNet architecture as an 155  
encoder, with ResNet-34 as the backbone in this study. The architecture incorporates residual connections, 156  
allowing gradient flow across layers without vanishing, which aids in learning deeper representations. This 157  
characteristic is particularly beneficial for segmentation tasks, as it ensures that the features extracted from 158  
deeper layers retain relevance to the task at hand. Pre-trained weights from ImageNet were utilized for 159  
initialization, enabling the model to benefit from a broad spectrum of features learned from natural images, 160  
which in turn helps improve the segmentation of synthetic hemothorax data. 161

**EfficientNet** EfficientNet (Tan & Le, 2020) represents a family of convolutional neural networks designed 162  
for better performance with fewer parameters. The scaling mechanism involves uniformly scaling depth, 163  
width, and resolution based on a compound coefficient. In this study, EfficientNet-B2 serves as the encoder 164

within the UNet framework, primarily for its computational efficiency and robust feature extraction. EfficientNet's depth-wise convolutions, combined with the batch normalization and Swish activation functions, contribute to the model's ability to capture complex thoracic features while maintaining reasonable training times and memory requirements.

### 3.1.3 Losses & Metrics

#### Dice Loss

Dice Loss (Sudre et al., 2017) is based on the Dice Coefficient, which measures the overlap between the predicted segmentation mask and the ground truth. It is defined as:

$$\text{Dice Loss} = 1 - \frac{2 \times |P \cap T|}{|P| + |T|}$$

where  $P$  represents the set of predicted pixels and  $T$  denotes the set of true pixels. This formulation directly optimizes for overlap, making it suitable for tasks where accurate boundary detection is critical. By focusing on maximizing the overlap, Dice Loss aids in the precise segmentation of the regions of interest.

#### Jaccard Loss

Jaccard Loss is derived from the Jaccard Index, a metric used to quantify the similarity between two sets. It is mathematically represented as:

$$\text{Jaccard Loss} = 1 - \frac{|P \cap T|}{|P \cup T|}$$

where  $P$  and  $T$  represent the predicted and true pixel sets, respectively. Jaccard Loss emphasizes the accuracy of the segmentation by evaluating the intersection relative to the union of the two sets, making it a suitable choice for assessing segmentation consistency.

#### IoU

The Intersection over Union (IoU) metric measures the overlap between predicted and true regions in segmentation tasks. It is calculated as:

$$\text{IoU} = \frac{|P \cap T|}{|P \cup T|}$$

where  $P$  and  $T$  correspond to the predicted and true segmentation masks, respectively. IoU serves as a primary evaluation metric in this study, providing a clear indication of the model's performance by quantifying the ratio of correctly predicted pixels to the total number of relevant pixels.

#### Binary Cross Entropy With Logits Loss

Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) is a commonly used loss function for binary classification tasks, particularly effective in segmentation models where each pixel is treated as a separate binary classification problem. It combines the sigmoid activation function and binary cross-entropy loss in a single step, making it numerically stable and efficient. The formulation is:

$$\text{BCEWithLogitsLoss} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\sigma(x_i)) + (1 - y_i) \cdot \log(1 - \sigma(x_i))]$$

where: -  $N$  is the number of pixels, -  $y_i$  is the ground truth label for pixel  $i$ , -  $x_i$  is the predicted logit for pixel  $i$ , -  $\sigma(x_i)$  represents the sigmoid function applied to the logits.

This loss function ensures accurate classification by transforming logits into probabilities in the range [0, 1] using the sigmoid function before calculating the cross-entropy. By handling raw logits directly, it avoids numerical instability, which can occur when using separate sigmoid and binary cross-entropy operations.

## Focal Loss

Focal Loss (Lin et al., 2018) is a modification of the standard cross-entropy loss, designed to address class imbalance by down-weighting the contribution of well-classified examples and focusing more on hard-to-classify instances. It introduces a modulating factor to the cross-entropy loss that reduces the loss for correctly classified pixels, thereby prioritizing learning from harder examples. The formulation is:

$$\text{Focal Loss} = -\frac{1}{N} \sum_{i=1}^N [\alpha \cdot (1 - p_i)^\gamma \cdot y_i \cdot \log(p_i) + (1 - \alpha) \cdot p_i^\gamma \cdot (1 - y_i) \cdot \log(1 - p_i)]$$

where: -  $N$  is the number of pixels, -  $y_i$  is the ground truth label for pixel  $i$ , -  $p_i$  is the predicted probability for pixel  $i$ , -  $\alpha$  is a weighting factor to balance positive and negative classes, -  $\gamma$  is the focusing parameter, controlling the rate at which easy examples are down-weighted.

The term  $(1 - p_i)^\gamma$  acts as a modulating factor, reducing the weight assigned to well-classified examples (where  $p_i$  is close to 1 for positive labels and close to 0 for negative labels). By doing so, Focal Loss concentrates the model's learning on challenging pixels, making it particularly suitable for tasks with severe class imbalance, such as medical image segmentation. The parameters  $\alpha$  and  $\gamma$  can be adjusted to control the balance between classes and the focus on hard examples.

## 3.2 Lung, Heart and Trachea Segmentation

Organs and important parts of thorax is segmented for bounding the thorax area. Bounding the area provides many benefits, such as estimating total volume of thoracic cavity and obtaining a boundary mask for limiting added synthetic objects.

### 3.2.1 Data & Preprocessing

The datasets used for lung, heart, and trachea segmentation are available on Kaggle (Konya, 2021). The dataset consists of 16,708 slices of lung region with corresponding masks. Images were resized to 512x512 and normalized. 0.8 to 0.2 ratio was used for train and validation sets.

### 3.2.2 Model & Training

A Unet with encoder of EfficientNet-B2, with weights trained on ImageNet dataset were used. Model takes data as a grayscale image, and outputs a 3 channel grayscale image. Each channel corresponds to one feature: lung, heart and trachea. Augmentations such as rotation and scaling were applied during training. Sigmoid function is applied to get probabilities from model output, and by selecting the probabilities with more than the threshold of 0.5, a binary mask is acquired. 5 fold cross validation was used.

## 3.3 Hemothorax Segmentation & Volume Estimation

### 3.3.1 Data & Preprocessing

With the synthetic data framework, objects with random parameters were inserted in the DICOM files of the patients. Series with artificial objects were converted to .png. Out of 5250 images, 1063 had at least one artificial object added.



<b>3.3.2 Model &amp; Training</b>	230
For training, a Unet model with ResNet34 as the encoder is used. Pretrained weights from ImageNet dataset	231
for encoder is used. For loss function, BCEWithLogits Loss and Dice Loss are used combined, with equal	232
weights. Adam is selected as the optimizer. During training, augmentations were applied such as scaling	233
and contrasting. 5 fold cross validation was completed. Testing was done with unseen synthetic images.	234
<b>3.3.3 Volume Estimation</b>	235
To estimate the volume of the hemothorax, a separate linear regression model is used for a rough volume	236
per pixel ratio. This data is obtained by running the artificial hemothorax code multiple times. Because	237
that the actual volume of the object added is known, this provides a sufficient ground truth. This must be	238
done only once at the beginning for each type of machine. The coefficients can be used for any scan after	239
the process is done.	240
<b>3.4 Pneumothorax Segmentation</b>	241
<b>3.4.1 Data &amp; Preprocessing</b>	242
For pneumothorax segmentation, the SIIM-ACR Pneumothorax Segmentation Competition data was used	243
(Zawacki et al., 2019). The dataset has 10,675 X-Ray scans after preprocessing, annotated by medical experts	244
from Society for Imaging Informatics in Medicine (SIIM).	245
<b>3.4.2 Model &amp; Training</b>	246
For training, a Unet model with ResNet50 as the encoder is used. Pretrained weights from ImageNet dataset	247
for encoder is used. For loss function, BCEWithLogits Loss, Dice Loss and Focal Loss were inspected, but	248
Combo loss was used. Adam is selected as the optimizer. During training, augmentations were applied such	249
as scaling and contrasting. 5 fold cross validation was completed. 0.8 to 0.2 ratio was used for training	250
and validation. Several augmentation techniques were applied during training, such as: flipping, random	251
contrast, gamma or brightness and scaling.	252
<b>4. Results</b>	253
<b>4.1 Synthetic Hemothorax Data Framework</b>	254
This framework processes selected DICOM series and integrates the synthetic objects based on user-defined	255
parameters. Volume calculations are performed to ensure the accuracy and consistency of the generated	256
objects.	257
The Monte Carlo method for volume estimation was tested on various geometric shapes and under	258
different conditions to assess its accuracy and consistency. The results demonstrate high precision across	259
different object types and sampling rates.	260
<b>4.1.1 Volume Estimation Accuracy</b>	261
Table 1 presents the comparison between actual volumes and those estimated using the Monte Carlo method	262
with 1 million samples for different geometric shapes.	263
The results show excellent agreement between the actual and estimated volumes, with errors consistently	264
below 0.20%. This demonstrates the high accuracy of the Monte Carlo method across different geometric	265
shapes.	266



Table 1: Accuracy of Monte Carlo Volume Estimation

Shape	Kwargs	Actual Volume	Estimated Volume ( $\text{mm}^3$ )	Error (%)
Cube	$h,l,w = 20,20,20$	8000	8000	0
Cylinder	$r,h = 5,15$	1178.098	1178.365	0.022
Blob	$A, \sigma_z, \sigma_y, \sigma_x, t = 1, 10, 15, 20, 0.2$	72571.517	72723.097	0.209
Sphere	$r = 10$	4188.79	4192.008	0.077
Rectangular Prism	$h,l,w = 10,20,30$	6000	6000	0

#### 4.1.2 Convergence Analysis

To assess the convergence of the Monte Carlo method, we performed volume estimations with increasing numbers of samples for an blob shape, with a base volume of **25.132741 $\text{mm}^3$** 's. Table 2 shows the results.

Table 2: Convergence of Monte Carlo Volume Estimation

Samples (n)	Volume ( $\text{mm}^3$ )	Error (%)
10,000	24.8256	1.2220
100,000	25.1356	0.0111
1,000,000	25.1172	0.0618
10,000,000	25.133357	0.0024

The estimated volume converges to the actual volume (25.132741) as the number of samples increases, with the estimate at 10 million samples differing by only 0.0024% from the actual value.

#### 4.1.3 Rotational and Scaling Invariance

We tested the method's invariance to rotation and scaling:

- **Rotation:** The volume of a rotated object differed from the original volume by only 0.19% (25.181376  $\text{mm}^3$ , 25.1328  $\text{mm}^3$ ).
- **Scaling:** When the object was scaled by a factor of 2, the estimated volume (200.552832  $\text{mm}^3$ ) was within 0.25% of the expected volume (201.062400  $\text{mm}^3$ ).

These results demonstrate that the Monte Carlo method maintains its accuracy under rotational and scaling transformations, which is crucial for real-world applications where objects may be oriented or sized differently across images.

In summary, the Monte Carlo method for volume estimation shows high accuracy, consistent convergence, and invariance to geometric transformations for various objects, making it a robust tool for volumetric analysis in medical imaging applications.

#### 4.1.4 Volume Estimation

Our results show that volume estimation was validated through test functions, and confirm that the Monte Carlo method used provides accurate and consistent results.

#### 4.1.5 Image Comparison

To illustrate the effectiveness of our framework in integrating synthetic objects into CT images, we present a comparison between original and modified images. Figure 1 shows side-by-side comparisons of original CT scans and the same scans with synthetic objects added.

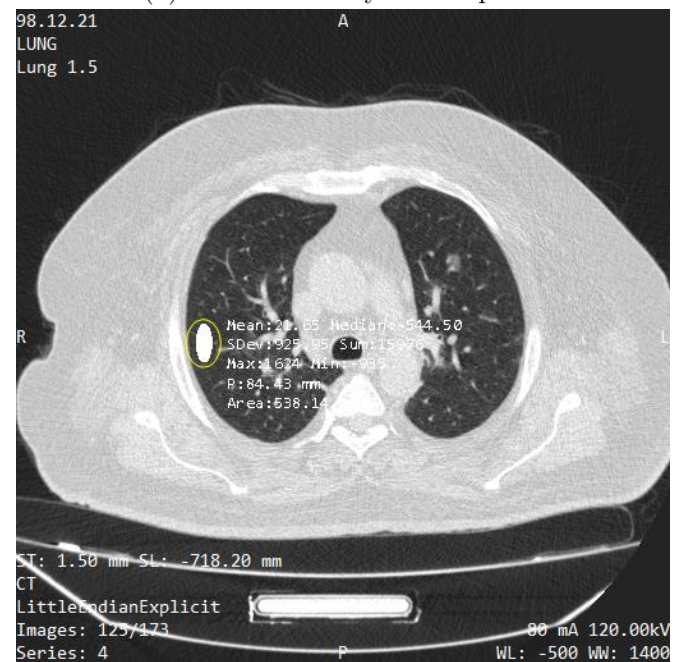
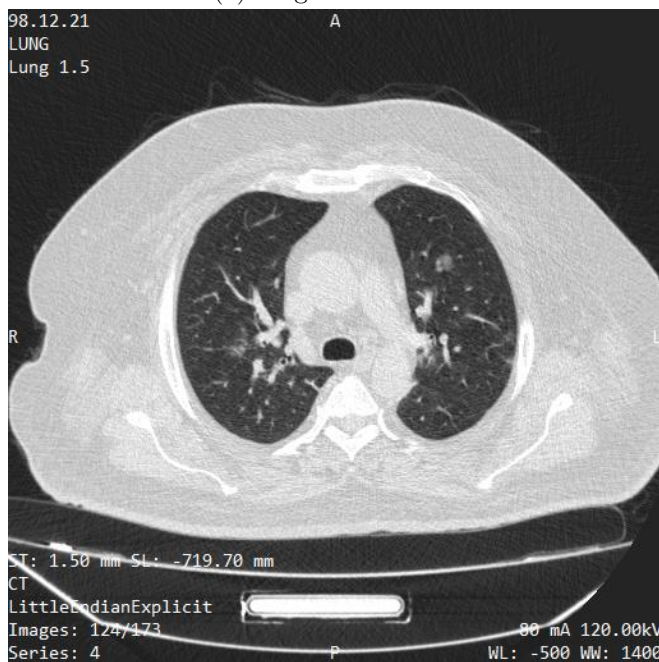
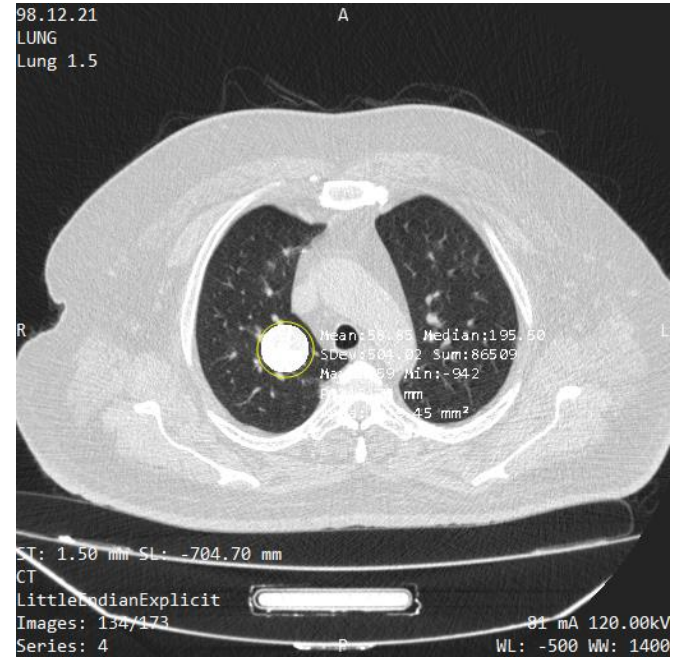
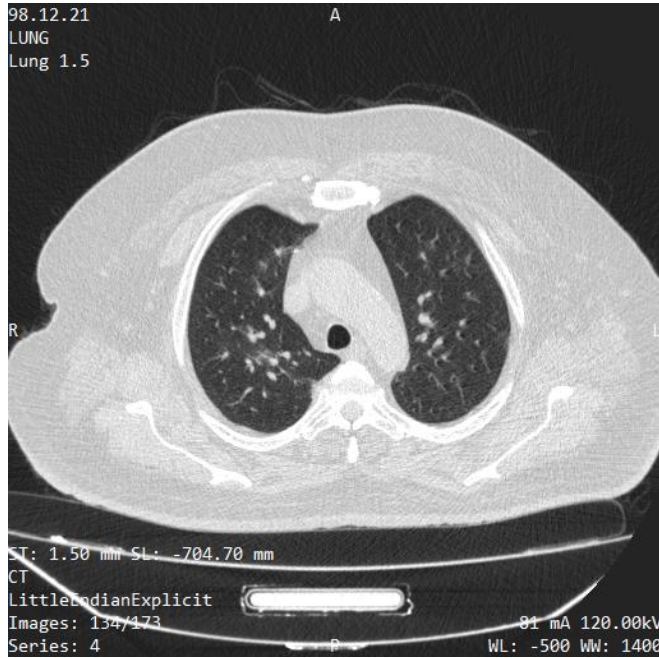


Figure 1: Comparison of original CT scans and scans with integrated synthetic objects

These images demonstrate the seamless integration of synthetic objects into the CT scans. The added 291 objects (a sphere in Figure 1b and an irregular blob (ellipsoid) in Figure 1d) are clearly visible and realistically 292 incorporated into the thoracic cavity, showcasing the framework's capability to augment medical imaging 293 data for various analytical and clinical applications with volumetric analysis incorporated. 294

## 4.2 Deep Learning 295

### 4.2.1 Lung, Heart and Trachea 296

The training was done on a Nvidia (Santa Clara, California, U.S.) P100 GPU 16GB, on Kaggle (San Francisco, 297 California, U.S.) platform. Model was trained for 70 epochs. A 0.8 to 0.2 ratio was used for train and 298 validation datasets. 299

#### Training Results 300

Train loss was 0.0152, and validation loss was 0.0233. Train dice score was 0.9622, and validation dice was 301 0.9537. Train and validation Jaccardi Index scores were 0.9391 and 0.9322 respectively (Jaccard, 1901). F1 302 scores for lung, trachea and heart are 0.9165, 0.7657 and 0.6458 respectively. The results can be viewed at 303 2. Example outputs can be viewed at project page. Confusion matrix can be viewed at 4a. 304

### 4.2.2 Hemothorax Segmentation & Volume Calculation 305

The training was conducted on an NVIDIA P100 GPU (16GB), manufactured by NVIDIA Corporation 306 (Santa Clara, California, USA) using the Kaggle platform, which is operated by Google LLC (Mountain 307 View, California, USA). Model was trained for 20 epochs. 0.9 to 0.1 ratio was used for positive and negative 308 object images. 0.8 to 0.2 ratio was used for train and validation datasets. 309

#### Segmentation Training Results 310

Results can be seen on the table, average of the 5 runs for each fold. Validation IoU was 0.84 at the end, 311 and loss was 0.0220. No signs of overfitting was observed. The results can be viewed at 3. Example outputs 312 can be viewed at project page. Confusion matrix can be viewed at 4b. F1 score of 0.98 was achieved. 313

#### Volume Calculation 314

Linear regression was done on 30 data points, and as hypothesized an R score of 0.999 and  $r^2$  of 0.998 was 315 achieved. 316

### 4.2.3 Pneumothorax 317

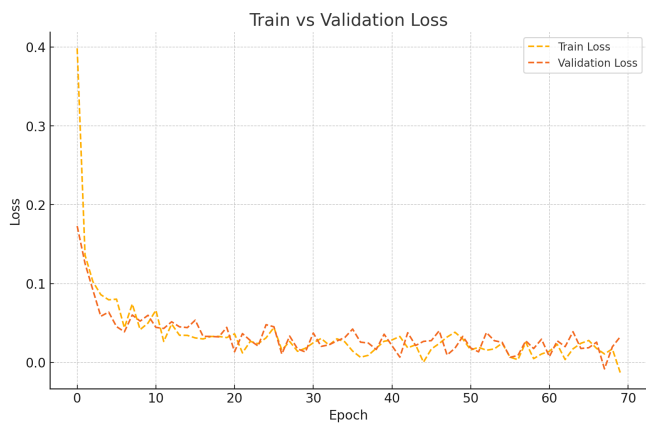
The training was done on a Nvidia P100 GPU 16GB, on Kaggle platform. Model was trained for 20 epochs. 318 0.8 to 0.2 ratio was used for train and validation datasets. 319

#### Training Results 320

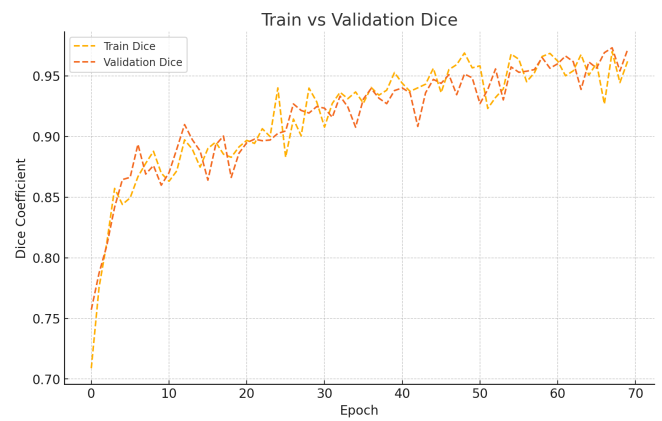
In average of 5 folds, a Val Dice score of 0.294, loss of 0.77 were achieved. Example outputs can be viewed 321 at project page. The F1 score at the end of the training is 0.36. Confusion matrix can be viewed at 4c. 322

## 5. Discussion 323

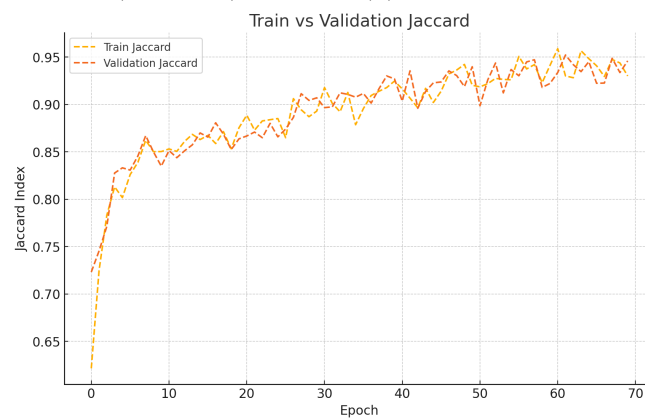
This study showed that synthetic objects could be used to develop and internally validation a model for 324 assessment of hemothorax, as well as other intra-thoracic lung pathologies. This method could be used for 325 development of other volumetric assessment tools with applications in health care. 326



(a) IoU of Training vs Validation (Averaged)

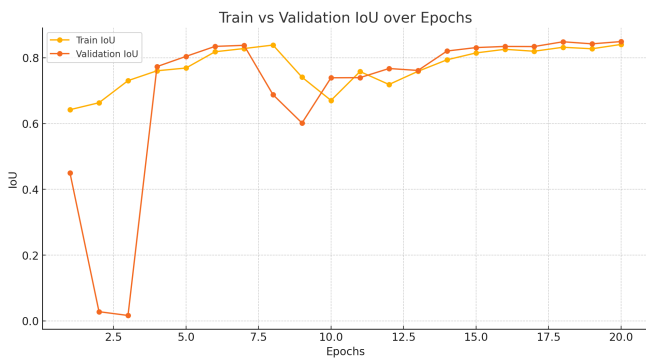


(b) Combined Loss of Training vs Validation (Averaged)

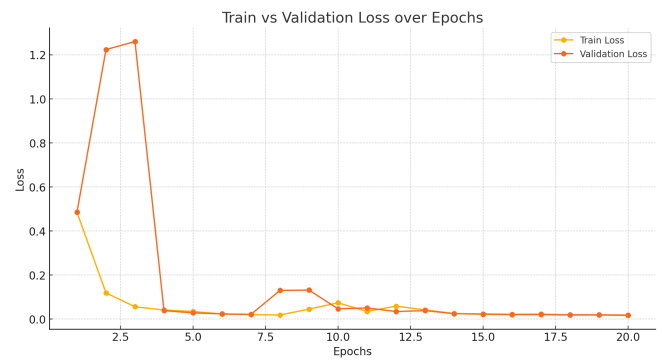


(c) Combined Loss of Training vs Validation (Averaged)

Figure 2: Results of chest model



(a) IoU of Training vs Validation (Averaged)

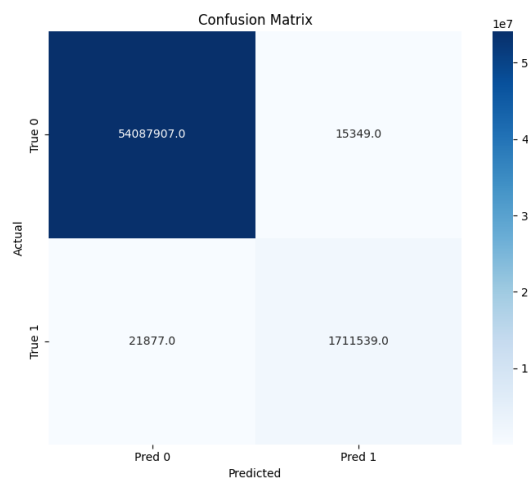


(b) Combined Loss of Training vs Validation (Averaged)

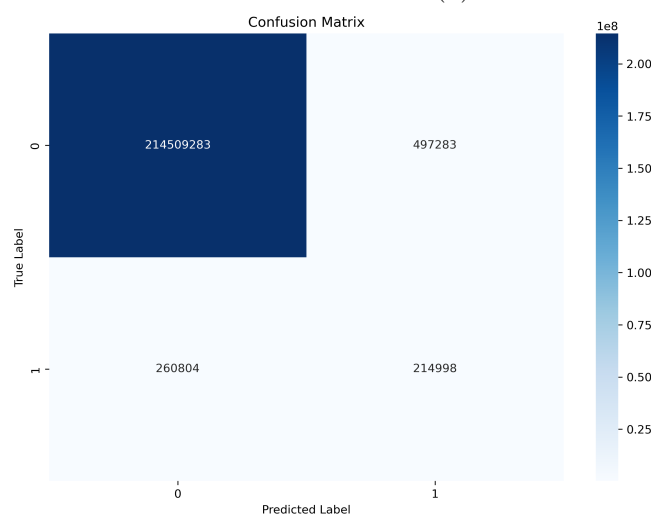
Figure 3: Results of hemothorax model



(a) Confusion matrix of chest model



(b) Confusion matrix of hemothorax model



(c) Confusion matrix of pneumothorax model

Figure 4: Confusion matrixes

The integration of synthetic objects into CT imaging data enhances clinical decision making, and can provide a robust tool for data augmentation and analysis, with volumetric assessment. This framework may be able to support various applications in medical imaging and related fields, helping clinicians and surgeons make key clinical decisions.

CT volumetry has been used for the assessment of a variety of pathophysiologies including in the intraabdominal cavity (Planz et al., 2019) as well as the thorax (Nair et al., 2023). However, it has not to our knowledge been used in comparison to determine greater clinical implication compared to other radiographic measures in thoracic trauma imaging.

The framework’s potential is demonstrated using synthetic data created for a CT hemothorax segmentation model. Also, the addition of classification of pneumothorax and the classification of other organ within the mediastinum point to the role of combined use of tools together to create a robust solution for medical analysis.

Next steps include validation of this framework for non-synthetic objects for volumetric assessment in the thorax, and testing hemothorax model with actual data or other synthetic data solutions such as GenerateCT (Hamamci et al., 2024). Further validation will be done externally, testing the model on hospital patients receiving video-assisted thoracoscopy or tube thoracostomy.

Limitations of the study include the lack of external validation of this model, as well as the lack of confirmation of a gold standard for intra-thoracic blood volume. Additional validation could be completed based on comparison of model with the amount of blood volume returned post-procedurally.

## 6. Conclusion

This study presents an initial framework for the integration and analysis of synthetic objects in medical imaging data. The methods and results demonstrate the potential for significant advancements in research and clinical applications.

Additionally, results of UNet models for various pathologies inside thoracic cavity are presented. A potential use case of the framework was exhibited. A possible approach for measuring volume of an object from CT imaging is shown.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Author Contributions

**First Author:** Conceptualization, Data curation, Methodology, Software. **Second Author:** Conceptualization, Methodology, Writing—original draft.

## Funding

No external funding was required for this study.

## Data Availability

Data supporting the findings of this study are available from the corresponding author upon reasonable request. The code for this analysis is available at: [GitHub Repository](#).



## Acknowledgments 363

The authors thank the institutions and individuals who contributed to the preparation of this manuscript, 364  
as well as the feedback from Dr. Brian Kim and Dr. Daniel Stephens at the Mayo Clinic, Department of 365  
Surgery. 366

## References 367

- Hamamci, I. E., Er, S., Sekuboyina, A., Simsar, E., Tezcan, A., Simsek, A. G., Esirgun, S. N., Almas, F., 368  
Dogan, I., Dasdelen, M. F., Prabhakar, C., Reynaud, H., Pati, S., Bluethgen, C., Ozdemir, M. K., 369  
& Menze, B. (2024). Generatect: Text-conditional generation of 3d chest ct volumes. <https://arxiv.org/abs/2305.16037> 370  
<https://arxiv.org/abs/2305.16037> 371
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. <https://arxiv.org/abs/1512.03385> 372  
<https://arxiv.org/abs/1512.03385> 373
- Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. 374  
*Bulletin del la Société Vaudoise des Sciences Naturelles*, 37, 547–579. 375
- Konya, S. (2021). Ct lung, heart, trachea segmentation [Accessed: 2024-10-24]. <https://www.kaggle.com/sandorkonya/ct-lung-heart-trachea-segmentation> 376  
<https://www.kaggle.com/sandorkonya/ct-lung-heart-trachea-segmentation> 377
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection. <https://arxiv.org/abs/1708.02002> 378  
<https://arxiv.org/abs/1708.02002> 379
- Mostafavi, S. M. (2021). Covid19-ct-dataset: An open-access chest ct image repository of 1000+ patients 380  
with confirmed covid-19 diagnosis [Harvard Dataverse, V1]. <https://doi.org/10.7910/DVN/6ACUZJ> 381
- Nair, A., Dyer, D., Heuvelmans, M., Mashar, M., Silva, M., & Hammer, M. (2023). Contextualizing the role 382  
of volumetric analysis in pulmonary nodule assessment: Ajr expert panel narrative review. *AJR Am* 383  
*J Roentgenol*, 220(3), 314–329. <https://doi.org/10.2214/AJR.22.27830> 384
- Planz, V., Lubner, M., & Pickhardt, P. (2019). Volumetric analysis at abdominal ct: Oncologic and non- 385  
oncologic applications. *Br J Radiol*, 92(1095), 20180631. <https://doi.org/10.1259/bjr.20180631> 386
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image seg- 387  
mentation. <https://arxiv.org/abs/1505.04597> 388
- Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as 389  
a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image* 390  
*analysis and multimodal learning for clinical decision support* (pp. 240–248). Springer International 391  
Publishing. [https://doi.org/10.1007/978-3-319-67558-9\\_28](https://doi.org/10.1007/978-3-319-67558-9_28) 392
- Tan, M., & Le, Q. V. (2020). Efficientnet: Rethinking model scaling for convolutional neural networks. 393  
<https://arxiv.org/abs/1905.11946> 394
- Zawacki, A., Wu, C., Shih, G., Elliott, J., Fomitchev, M., Hussain, M., Lakhani, P., Culliton, P., & Bao, S. 395  
(2019). Siim-acr pneumothorax segmentation [Kaggle]. [https://kaggle.com/competitions/siim-acr-](https://kaggle.com/competitions/siim-acr-pneumothorax-segmentation) 396  
[pneumothorax-segmentation](https://kaggle.com/competitions/siim-acr-pneumothorax-segmentation) 397