Categorizing deep nnUNet-based segmentation errors across organs from abdominal CT volumes

Progress	0%
Priority	High
Assignee	Ⓢ Surya
Status	In progress

▼ **[Protocol] Categorizing deep learning-based 3D segmentation errors across organs from abdominal CT volumes**

▼ **Abstract**

There is a dearth of literature that systematically benchmarks organs for the kinds of errors that AI-based segmentation models are likely to make while performing segmentation on CT scans. Consequently, we lack understanding of image features that lead to specific kinds of AI segmentation errors. Metrics that define segmentation quality such as Dice similarity coefficient (DSC) or intersection over union (IoU), do not tell us what kind of error the model is making — only the extent of the error made. The kinds of error that AI-based models make while segmenting each organ is essential for us to understand if we were to design models that avoid these errors. The purpose of this work is to characterize the errors made by popular, AI-based organ segmentation models. We formulate a set of quantitative metrics that indicates the degree and kind of errors made by AI models, and subsequently identify organ-specific features that are susceptible to such errors. Lastly, we analyze localization statistics on organs in an effort to correlate organ types and characteristics with the regions that are most susceptible to segmentation errors.

▼ **1. Introduction**

▼ While several benchmarking studies exist, which categorize organs based on difficulty of segmentation by AI-based models...

▼ **Categorization of organs based on difficulty of segmentation by AI-based models, from abdominal CT scans**

Several benchmarking studies and literature reviews have categorized organs based on difficulty of segmentation by AI-based models.

▼ **Benchmarking paper 1 [1]**:

In Table L1 **(see Table L1 in Section x. Literature Review/ Tables)**, the authors of [1] have provided a comparison among five organs, namely the liver, spleen, right and left kidneys, and the pancreas. As can be observed from the table, 3D nnU-Net performs much better in terms of Dice score on the liver, spleen and kidneys as compared to the pancreas, for both the SECT single-energy photon CT modality and the DECT double-energy photon modality, for both the portal venous phase contrast type and the virtual non-contrast settings. In the same work [1], the authors also used **"agreement"** between volume of actual organ (ground truth) and that segmented by the AI segmentation model (predicted) — where **"agreement"** was calculated using **intraclass correlation coefficients (ICCs)** — as a proxy for segmentation quality. As can be seen from Table L2 **(see Table L2 in Section x. Literature Review/ Tables)**, the agreement (ICC) between ground truth organ segmentation and that by the 3D nnU-Net model, is extremely high for liver, spleen and kidneys (ICC>0.90 across all contrast types), as compared to the pancreas (ICC>0.79 across all contrast types).

▼ **Benchmarking paper 2 [2]**:

The authors of [2] acquired a large abdominal CT dataset (**150 CT volumes** — containing **30,500 slices**), and subsequently performed organ-wise (**16 organs**) benchmarking of AI-based segmentation model performance. Table L3 **(see Table L3 in Section x. Literature Review/ Tables)** provides DSCs for all 16 organs.

As can be observed from Table L3, the **organs with high (DSC>85%) segmentation scores** across all of the 10 popular AI-based segmentation models are, in descending order:

1. Liver
2. Heads of femur (average of Left and Right Femur DSC)
3. Spleen
4. Bladder
5. Kidneys
6. Stomach

From the same Table L3, organs with **moderate segmentation scores (75% <DSC< 85%)** are, in descending order:

1. Colon
2. Pancreas
3. Rectum
4. Gallbladder
5. Esophagus

From the same Table L3, organs with poor segmentation scores (DSC<75%) are, in descending order:

1. Adrenal gland
2. Duodenum

The authors of [2] also provide a table (Table L4 — **see Table L4 in Section x. Literature Review/ Tables)** with mean difference in DSC between a senior radiologist (7+ years of experience) and an expert radiologist (20+ years of experience), and the consensus segmentation.

From Table L4, organs with high segmentation (DSC difference between expert radiologist and consensus < 2.5 **AND** DSC difference between senior radiologist and consensus < 1.4), in descending order:

1. Kidneys (mean of L and R DSCs)
2. Spleen
3. Heads of femur (mean of L and R DSCs)
4. Liver

From Table L4, organs with medium segmentation (DSC difference between expert radiologist and consensus between 2.5 and 5.0 **while** DSC difference between senior radiologist and consensus stays around $1.3 - 1.5$), in descending order:

1. Bladder
2. Intestine
3. Rectum
4. Stomach
5. Colon

From Table L4, organs with low segmentation (DSC difference between expert radiologist and consensus between > 5.0 **AND** DSC difference between senior radiologist and consensus > 2.5), in descending order:

1. Gallbladder
2. Pancreas
3. Esophagus
4. Duodenum
5. Adrenal gland

Note : As expected there is a high positive correlation between the difference in DSC between expert and consensus, and that between the senior and the consensus. Therefore, selection above is done based on the DSC difference of the expert, since the expert's DSC difference has higher variance, therefore is easier to distinguish. For example, the senior radiologist's DSC difference from consensus is barely any different between the high and the medium segmentation — there is some observable difference between high and low segmentation quality for senior radiologist.

In [2], 3 junior radiologists (3 years of experience) we asked to improve the results of nnUNetV2(3D) — the results can be found in Fig L1.

Organs in descending order of segmentation performance:

- $DSC > 95.0$
  - Liver
  - Spleen
  - Kidneys
  - Bladder
- $90.0\% < DSC < 95.0\%$
  - Stomach
  - Head of femur (mean of L and R DSCs)
- $85.0\% < DSC < 90.0\%$
  - Colon
  - Intestine
- $80.0\% < DSC < 85.0\%$
  - Pancreas
  - Esophagus
- $75.0\% < DSC < 80.0\%$
  - Rectum
- $70.0\% < DSC < 75.0\%$
  - Gallbladder
- $DSC < 70.0\%$
  - Duodenum
  - Adrenal gland

Observation from Fig L1 in [2]:

Certain organs perform really poorly when segmented by AI-based segmentation models. These organs, after manual correction, still remain poorly segmented as compared to organs that were well segmented by the model in the first place — which indicates that organs that are poorly segmented by AI are also likely to be poorly segmented by humans.

→ I want to investigate: What features make them perform better after human correction. [Explained later, in Methods.]

▼ **Official AMOS [3] abdominal CT leaderboards**:

▼ **First, a little bit about the AMOS dataset**:

- No. of volumes: 600
  - 500 CT scans
  - 100 MRI scans
- Organs: 15
  - Aorta
  - Duodenum
  - Esophagus
  - Gallbladder
  - Inferior vena cava
  - Left adrenal gland
  - Right adrenal gland
  - Left kidney
  - Right kidney
  - Liver
  - Pancreas
  - Reproductive organs
    - Male: prostate
    - Female: uterus
  - Spleen
  - Stomach
  - Urinary bladder

▼ **AMOS CT Regular Evaluation (Test) Leaderboard [access leaderboard link here]**:

▼ **Top 3:**

▼ Organ-wise DSCs for AI-based segmentation model **(ordered by decreasing order of DSC)**:

Rank 1	Rank 2	Rank 3
liver 0.9817	**liver** 0.9821	**liver** 0.9811

spleen 0.9769	**spleen** 0.9764	**spleen** 0.9762
left kidney 0.9733	**left kidney** 0.9729	**left kidney** 0.9733
right kidney 0.9707	**right kidney** 0.9702	**right kidney** 0.9703
aorta 0.9641	**aorta** 0.9623	**aorta** 0.9629
stomach 0.9499	**stomach** 0.9509	**stomach** 0.9491
postcava 0.9341	**postcava** 0.9334	**postcava** 0.9336
bladder 0.9309	**bladder** 0.9322	**bladder** 0.9258
pancreas 0.9092	**pancreas** 0.9072	**pancreas** 0.9078
gallbladder 0.9050	**esophagus** 0.8903	**esophagus** 0.8906
esophagus 0.8940	**gallbladder** 0.8850	**gallbladder** 0.8819
duodenum 0.8812	**duodenum** 0.8786	**duodenum** 0.8743
prostate/uterus 0.8552	**prostate/uterus** 0.8533	**prostate/uterus** 0.8536
left adrenal gland 0.8362	**left adrenal gland** 0.8344	**left adrenal gland** 0.8323
right adrenal gland 0.8088	**right adrenal gland** 0.8063	**right adrenal gland** 0.8049

▼ Organ-wise NSDs for AI-based segmentation model **(ordered by decreasing order of NSD)**:

Rank 1	Rank 2	Rank 3
spleen 0.9305	**spleen** 0.9255	**spleen** 0.9257
aorta 0.9266	**aorta** 0.9218	**aorta** 0.9233
left kidney 0.9184	**left kidney** 0.9173	**left kidney** 0.9176
right kidney 0.9169	**right kidney** 0.9147	**right kidney** 0.9146
liver 0.8765	**liver** 0.8743	**liver** 0.8721
left adrenal gland 0.8567	**left adrenal gland** 0.8521	**left adrenal gland** 0.8494
right adrenal gland 0.8485	**right adrenal gland** 0.8471	**right adrenal gland** 0.8422
postcava 0.8371	**postcava** 0.8347	**postcava** 0.8341

esophagus 0.8321	**stomach** 0.8253	**esophagus** 0.8274
stomach 0.8293	**esophagus** 0.8243	**stomach** 0.8249
gallbladder 0.8239	**bladder** 0.8242	**bladder** 0.8138
bladder 0.8230	**gallbladder** 0.8013	**gallbladder** 0.7999
pancreas 0.7740	**pancreas** 0.7701	**pancreas** 0.7707
duodenum 0.7623	**duodenum** 0.7557	**duodenum** 0.7513
prostate/uterus 0.6537	**prostate/uterus** 0.6529	**prostate/uterus** 0.6552

▼ **Other repositories:**

- **Access results link here.**

▼ **TotalSegmentator [4]**:

▼ **About the TotalSegmentator dataset**

▼ Training set

- No. of CT volumes = 1082 volumes (90%), randomly picked from routine clinical exams between 2012 - 2020.
- Anatomic structures = 104 (27 organs, 59 bones, 10 muscles and 8 vessels)

▼ Validation set

- No. of CT volumes = 57 (5%)

▼ Test set

- No. of CT volumes = 65 scans (5%)

▼ **About the TotalSegmentator study**

- An **nnU-Net** segmentation algorithm was trained and tested on the aforementioned TotalSegmentator dataset.
- I downloaded the json files with the complete set of per-organ DSC and NSD scores on their GitHub repository — and plotted the scores for organs commonly studied in abdominal CT volumes (spleen, liver, pancreas, esophagus, kidneys, adrenal glands, stomach, colon, bladder, gallbladder and aorta) in Table L5 and Table L6 **(see Table L5 and Table L6 in Section x. Literature Review/ Tables)**.
  - As is evident from Tables L5 and L6,

▼ **Organs that we are focusing on**

At least for the initial analyses, we focus on the organs fully contained within abdominal CT volumes, namely:

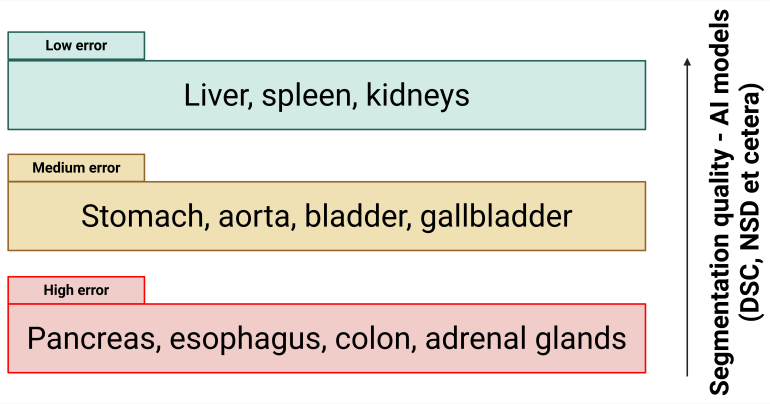
1. Spleen
2. Liver
3. Pancreas
4. Esophagus
5. Kidneys

6. Adrenal glands
7. Stomach
8. Colon
9. Bladder
10. Gallbladder
11. Aorta
12. ~~Small intestine~~
13. ~~Appendix~~
14. ~~Bile ducts~~
15. ~~Inferior vena cava~~
16. ~~Arteries and veins~~
    a. ~~Renal arteries~~
    b. ~~Renal veins~~
    c. ~~Portal vein~~
    d. ~~Mesenteric vessels~~
17. ~~Spine~~
18. ~~Lymph nodes~~
19. ~~Lower ribs~~

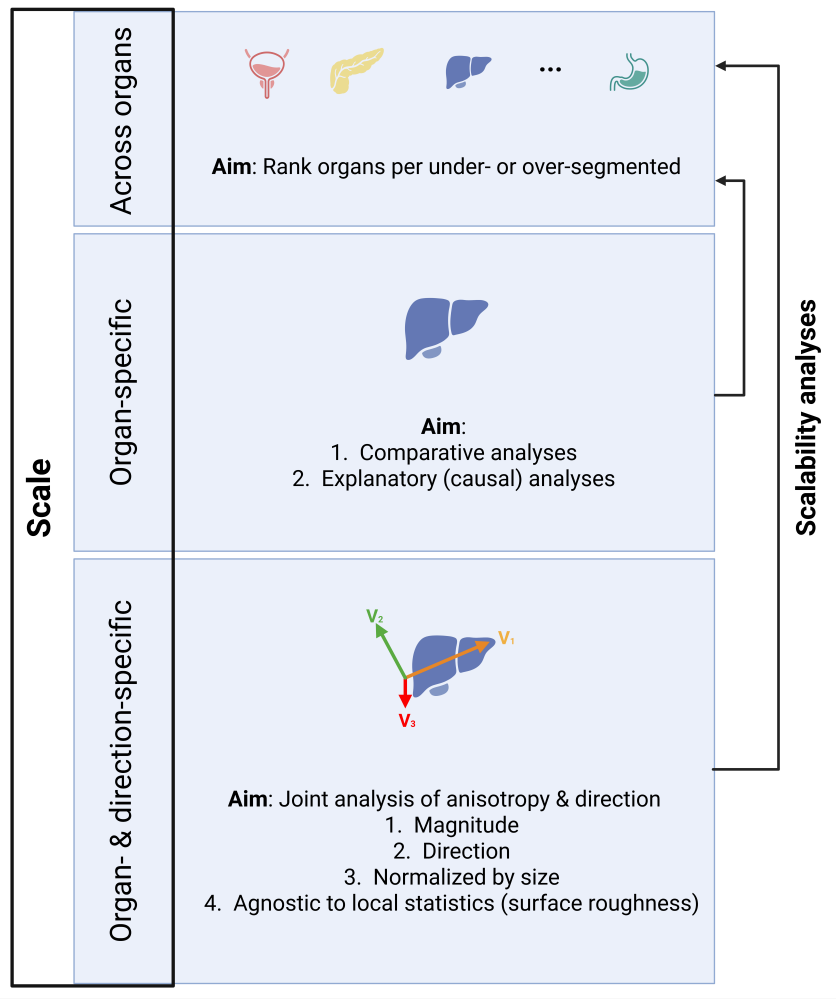
▼ Why **~~small intestine~~**, **~~appendix~~**, **~~bile ducts~~**, **~~inferior vena cava~~**, **~~arteries and veins~~**, **~~lymph nodes~~**, **~~spine~~** and **~~lower ribs~~** are not included?

- They are not “popular” organs, meaning I have not seen them being segmented in many studies.
  - Consequently, not all databases have these ground truths.
- Additionally, when it comes to the **~~small intestine~~** specifically, the end of the large intestine is difficult to differentiate from the beginning of the small intestine is. Additionally, some abdominal CT volumes contain the ending of the large intestine, while some have the beginning of the small intestine missing. It is just not a great organ to include in a segmentation error benchmarking study.
- Same with the **~~spine~~** — not a popular organ. Additionally, the whole spine is not even included in the CT volumes. Some databases (such as TotalSegmentator) have ground truths for vertebrae, but we are leaving those out as well.
- Leaving out the **~~lower ribs~~** as well — although TotalSegmentator has ground truths for them.

▼ My understanding of how well AI-based segmentation models perform on different organs, based on literature



▼ We are not just interested in quality of segmentation (good, moderate, bad) — we are more interested in the type of segmentation errors.



▼ CT datasets to be used in the study

▼ **Selection criteria**

We run these AI-based segmentation models on populat CT datasets that fulfill the following criteria:

1. Contain either all or a subset of the 11 aforementioned, popular (meaning, commonly studied in segmentation studies on CT datasets), abdominal organs that we are interested in:
    a. Spleen
    b. Liver
    c. Pancreas
    d. Esophagus
    e. Kidneys
    f. Adrenal glands
    g. Stomach
    h. Colon
    i. Bladder
    j. Gallbladder
    k. Aorta
2. Contain at least 30 volumes
3. All volumes in 3D — not 2D (meaning, having multiple 2D slices)
4. Have published an AI segmentation model on their data — or are popular enough. Meaning, published at a popular venue such as NeurIPS, CVPR, MICCAI, MIDL et cetera — and therefore have an extensive leaderboard.

    ⇒ This criteria will help us ensure that our AI model-based segmentation results actually match those of the state-of-the-art. And this is not for performance (we are not trying to implement a segmentation model with high DSC) — this is more meant as a sanity check. A popular database is just more credible — and does not require us to check whether the 3D organ ground truth is in fact correct.

▼ **Datasets to be used in the study**

We plan on using 6, 3D CT datasets that follow the criteria above:

▼ 1. The DECT/ SECT dataset [1]

▼ Contains two CT contrast types:

- **Dual-energy CT (DECT)**: 95, 3D volumes.

  In the original study [1], the dataset was split as follows:

  - Training set: 75 volumes
  - Validation set: 10 exams
  - Internal test set: 10 volumes
- **Single-energy CT (SECT)**: 30, 3D volumes.

  In the original study [1], the 30, 3D SECT volumes were usedas an indepedent, external test set to evakuate the performance of the AI-based segmmentation algorithms.

▼ Organs classes included:

- Liver
- Spleen
- Kidney(L)

- Kidney(R)
- Pancreas

▼ 2. WORD [2]

▼ The dataset

No. of CT volumes: 150

▼ 16 organ classes included:

- Liver
- Spleen
- Kidney(L)
- Kidney(R)
- Stomach
- Gallbladder
- Esophagus
- Pancreas
- Duodenum
- Colon
- Intestine
- Adrenal(L)
- Adrenal(R)
- Rectum
- Bladder
- Head of Femur(L)
- Head of Femur(R)

▼ 3. AMOS [3]

- No. of volumes: 600
  - 500 CT scans
  - 100 MRI scans
- Organs: 15
  - Aorta
  - Duodenum
  - Esophagus
  - Gallbladder
  - Inferior vena cava
  - Left adrenal gland
  - Right adrenal gland
  - Left kidney
  - Right kidney
  - Liver

- Pancreas
- Reproductive organs
  - Male: prostate
  - Female: uterus
- Spleen
- Stomach
- Urinary bladder

▼ 4. TotalSegmentator [4]

▼ **About the TotalSegmentator dataset**

▼ Training set

- No. of CT volumes = 1082 volumes (90%), randomly picked from routine clinical exams between 2012 - 2020.
- Anatomic structures = 104 (27 organs, 59 bones, 10 muscles and 8 vessels)

▼ Validation set

- No. of CT volumes = 57 (5%)

▼ Test set

- No. of CT volumes = 65 scans (5%)

▼ 5. BTCV [5]

▼ About the dataset

- No. of CT volumes = 47

▼ 13 organ classes:

- Aorta
- Esophagus
- Gallbladder
- Inferior vena cava
- Left adrenal gland
- Left kidney
- Liver
- Pancreas
- Portal vein and splenic vein
- Right adrenal gland
- Right kidney
- Spleen
- Stomach

▼ 6. FLARE [6]

▼ About the dataset

▼ Total no. of CT volumes = 2300

- Unlabeled (no ground truth available): 2000

▼ Labeled: 300

In the original work [6]:

- Training data: 50 volumes
- Validation data: 50 volumes
- Test data: 200 volumes

▼ **nnUNet-based segmentation models used in the study**

- nnUNet (2D)
- nnUNetV2 (2D)
- nnUNet (3D)
- nnUNetV2 (3D)
- TotalSegmentator

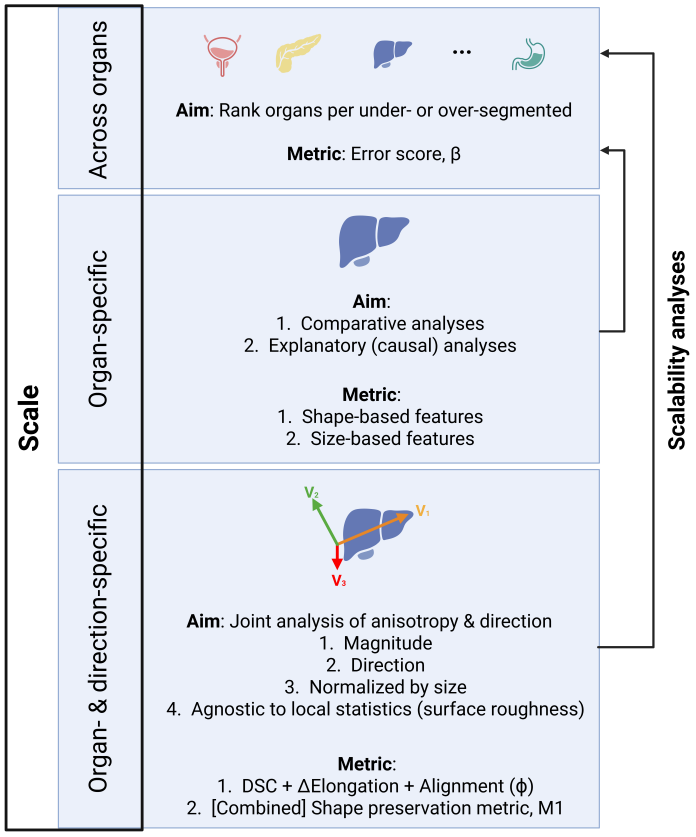
▼ 2. Materials and methods

▼ 2.1 Study overview

The goal of the study is to be able to characterize the errors produced by AI segmentation-based models on an organ-specific level such that going forward we could develop segmentation models that are more capable of avoiding such errors.

We first run the trained TotalSegmentator model, in addition to four other popular nnUNet-based segmentation segmentation models (namely nnUNet (2D), nnUNetV2 (2D), nnUNet (3D) and nnUNetV2 (3D)) — on 11 abdominal organs (spleen, liver, pancreas, esophagus, kidneys, adrenal glands, stomach, colon, bladder, gallbladder and aorta) — from the CT volumes in 6 databases (namely, DECT/ SECT dataset [1], WORD [2], AMOS [3], TotalSegmentator [4], BTCV [5] and FLARE [6]).

Then, we categorise the error on a per-organ basis, based on the most wide catgorization (over- and under-segmentation), and subsequently more fine-grained categorizations. Next, we correlate the characteristics (features) of organs with the kinds of errors that models make while segmenting them. Finally, we try to localize these errors on the part of the organ, in order to better understand not only what organ characteristics make what kinds of errors, but also where in the organ certain features encourage incorrect segmentation.



▼ 2.2 Methods

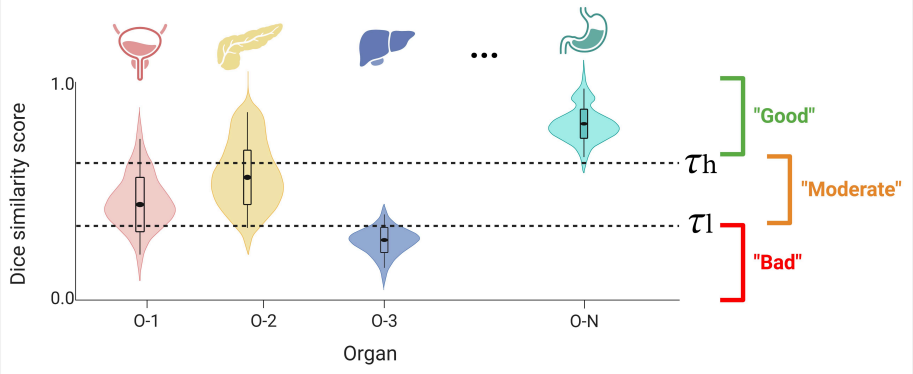
▼ 2.2.1 Ranking organs as per susceptibility to being under- or over-segmented

[Note: This is the most basic, crude level of analyses.]

In this study, we first perform segmentation on five organs (liver, kidney, esophagus, pancreas and inferior vena cava) from four abdominal CT segmentation databases [1-3], using the pre-trained TotalSegmentator model [4].

We expect there to be some variability in terms of segmentation quality (DSC, NSD, IoU), depending on the overall performance of AI segmentation models on individual organs:

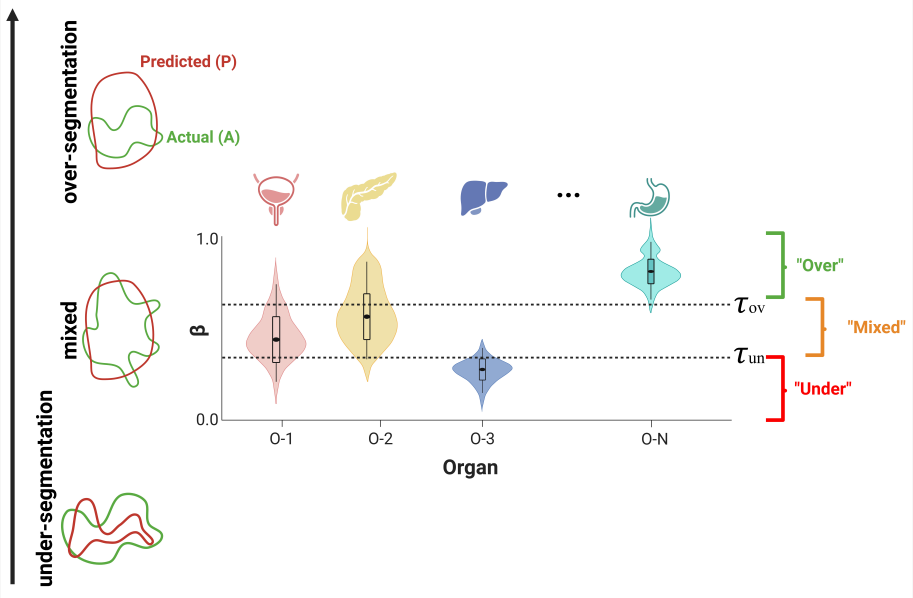
- Organs with median $DSC < \tau_1$: "Bad" segmentation quality
- Organs with median $\tau_l < DSC < \tau_h$: "Moderate" segmentation quality
- Organs with median $DSC > \tau_h$: "Good" segmentation quality



Many works have actually done this kind of analysis across organs, often focussing on certain demographics — for example [3] does this based on age and sex.

⇒ However, we are not just interested in quality of segmentation (good, moderate, bad) — we are more interested in the type of segmentation errors (over-segmentation, under-segmentation).

Instead, we do the gradation as follows:



Formulation of a deterministic and completely quantitative metric for measuring the extent of under- and over-segmentation

▼ **Goal — *What?***

Instead of **characterizing** the segmentations based on quality (good, moderate, bad), we are trying to find a metric that **quantifies** the segmentations based on error type (over-segmentation, under-segmentation).

▼ **Purpose — *Why?***

With a quantitative metric of error type, we can then perform correlation, regression et cetera to infer:

- What organ features are related to under-/ over-segmentation?
- What region in the organ is susceptible to be under-/ over-segmented?

⇒ Correlation- and/ or regression-based analyses are impossible with categorical dependent variables.

⇒ More importantly, we **already know** what organs are easier or more difficult to segment by AI segmentation models. We want to know **what kind** of error they are more likely to make on an organ-wise basis.

▼ **The mathematical metric that we shall use for quantifying under- and over-segmentation — *How?***

Error score, β, is defined as the difference between the area of the predicted segmentation and the ground truth segmentation, normalized by the area of the ground truth segmentation:

$$\beta = \frac{Area(predicted) - Area(actual)}{Area(actual)}$$

The error score, β, instead of characterizing the segmentation based on quality (good, moderate, bad), characterizes the segmentation based on error type (over-segmentation, under-segmentation).

▼ **Range of values of error score (β) — and what they mean [formal, mathematical quantification of segmentation error type]**

- $\beta > 0$: over-segmentation
- $\beta < 0$: under-segmentation
- $\beta = 0$: Either perfect segmentation, or just bad (mixed over- + under-) segmentation
  - $\beta = 0$ and low DSC: mixed over + under segmentation
  - $\beta = 0$ and high DSC: good segmentation

▼ **Rationale (conceptual justification) for aforementioned mathematical quantification of error score (β)**

▼ β is a **relative-area**, **signed** error metric.

What do the **sign** and **magnitude** of β signify?

- **Magnitude** of β, $|\beta|$: Signifies how big of an under-/ over-segmentation error it is, **relative to area of the organ** in question. → ***relative-area metric***
- **Sign** of β: Signifies type of error (under- or over-segmentation). → ***signed metric***

▼ **Advantages of β**

▼ β does not require additional scale normalization — as it is already normalized for organ size. **Therefore, β is comparable across different organs**.

- Let $Area(predicted) = A_p$ is k times as large as the true surface area of the organ $A(actual) = A_a$.
- In that case, \beta = k-1 = a constant.
- Irrespective of whether the organ is massive like the liver (large A_a) or tiny like the prostate (small A_a), $\beta = k - 1$ **ALWAYS** represents the surface area of AI model-based segmentation as being k times as large as the real area of the organ.

▼ **Shortcomings of β**

While \beta is unbiased to volume or area of the organ in question, it is not a perfect, stand-alone metric. Some of the challenges of error score (β) are as follows:

▼ **β is not a good metric for spatial overlap — and must be used in combination with DSC, NSD, IoU or other metrics that are spatially aware**.

- $\beta = 0$ is a special case, which represents that $A_p = A_a$.
- Therefore, must be used in combination with DSC, NSD, IoU or other spatially aware metrics.
- $\beta = 0$: Either perfect segmentation, or just bad (mixed over- + under-) segmentation
  - $\beta = 0$ and low DSC: mixed over + under segmentation
  - $\beta = 0$ and high DSC: good segmentation

⇒ **Note: We will circle back to this at the end of our analyses.**

▼ **2.2.2 Organ-wise characterization of more nuanced errors (errors beyond degree of under- and over-segmentation)**

▼ Types of analyses — **purely comparative** versus **explanatory/ causal****

Note: It is important to note the distinction between **the way that we use** the combination of **1. error features** and **2. organ features** during the course of our analyses:

▼ 1. **[Purely comparative]** For the purpose of error characterization

▼ **What question** are we trying to answer here?

How does the segmentation model **erroneously alter** the segmentation characteristics of the organ?

▼ Comparison criteria

This type of analysis (**purely comparative**) only occurs between the **same feature** of the ground truth and the predicted segmentation.

▼ Comparison techniques

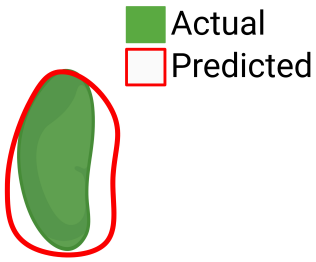
1. Correlation between the independent variable (feature F of the ground truth geometry) and the dependent variable (same feature F of the predicted geometry).

2. Simple linear regression:

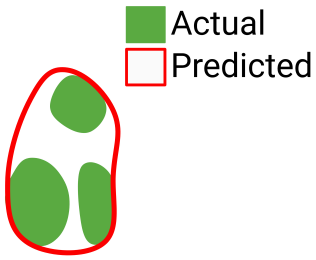
- Independent variable — same as above (feature F of the ground truth geometry)
- Dependent variable — same as above (feature F of the predicted geometry)

▼ Examples

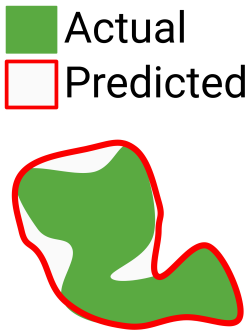
▼ Do AI segmentation models **tend to make** elongated structured more circular?



▼ Do AI segmentation models **tend to combine** small sub-structures into an all-encompassing bigger structure?



▼ Do AI segmentation models **tend to make** nuanced shapes seem more simplistic?



▼ 2. **[Explanatory]** For the purpose for inferring **what organ characteristics** lead to **what types of errors**

▼ **What question** are we trying to answer here?

What features lead to **what kinds of errors**?

Note: Error score β, is one of the F_y features, as we are also interested in what shape statistics might be causing under- or over-segmentation.

▼ Regression and/ or correlation criteria

This type of analysis (**explanatory**) occurs between **different features** extracted from the ground truth and the predicted segmentation.

⇒ In other words, are the shape-based features **causal****? Does shape feature F_x of the ground truth have an effect on error feature F_y that occurs between the ground truth and the prediction? **[As explained before, error score β, is one of the F_y features, as we are also interested in what shape statistics might be causing under- or over-segmentation.]**

*[****Note**: Correlation is used as a proxy for causation here.]*

▼ Examples

- Does a rectangular liver lead to under-segmentation as compared to a spherical liver?
- Is a complex shape more likely to be over-segmented?
- Is a complex shape more likely to be deformed (and is it in the direction of the concavity — if the complexity is a concavity) than a smoother shape?

▼ **Features of interest**

▼ **Shape-agnostic features (only 2D features in the entire analysis)**

▼ **Note**: These are the most naive, most simplistic features.

- Have nothing to do with shape — only concerned with **size**.
- We do not expect to receive any great insights from these features.
- Our only expectation is to find a minimum and/ or maximum range of organ sizes below/ above which the segmentation models perform poorly.
- **These are per-slice 2D features**.

▼ **What question** are we trying to answer with these features?

Is there a threshold on the minimum size of the organ under which the segmentation model just **ignores** the organ as noise?

▼ List of features

1. Area (2D)
2. Biggest radius completely inside the per-slice 2D geometry

▼ **Shape-based features**

▼ **Measures of anisotropy**

▼ **Note**

- It must be noted that these are features of anisotropy, **not surface roughness (surface irregularity)**.
- Anisotropy: The elongatedness (non-symmetry) of the 3D shape as a whole.
- Surface rough (surface irregularity): How rough the surface is.
- A 3D geometry can be any combination of isotropic/ anisotropic **and** rough.

▼ **Sphericity**

▼ **Definition**

How closely does a 3D shape resemble a sphere?

▼ **Mathematical formulation**

$sphericity(V, A) = \frac{\text{surface area of sphere with volume } V}{\text{surface area of the actual object } (A)} = \frac{(36\pi V^2)^{1/3}}{A}$,

where:

- V = volume of 3D shape in question
- A = surface area of 3D shape in question

▼ **Conceptualization and derivation of mathematical formulation**

▼ **Conceptualization**

Sphericity is a ratio of the minimum surface area possible for the volume of the object in question (which is, basically a sphere with the same volume) — divided by the actual surface area of the object in question.

▼ **Derivation of mathematical formulation**

$sphericity(s) = \frac{\text{surface area of sphere with volume } (V)}{\text{surface area of the actual object } (A)} \rightarrow (1)$

We know that, volume of a sphere with radius r is:

$V = \frac{4}{3}\pi r^3 \rightarrow (2)$

We know that surface area of a sphere with radius r is:

$A = 4\pi r^2 \rightarrow (3)$

From (2) and (3), the surface area of a sphere with volume V, is:

$A(V) = 4\pi(V^2) = 4\pi(\frac{4}{3}\pi r^3)^2 = (36\pi V^2)^{1/3} \rightarrow (4)$

Substituting A(V) from (4) in (1), we get:

$$sphericity(s) = \frac{\text{surface area of sphere with volume } (V)}{\text{surface area of the actual object } (A)} = \frac{(36\pi V^2)^{1/3}}{A}$$

▼ **Relative surface area excess (RSAE)**

▼ **Definition**

Ratio of **excessive** surface area of given 3D geometry, relative to that of a sphere having the same volume.

▼ **Mathematical formulation**

$$RSAE(V, A) = \frac{\text{surface area of the actual object } (A) - \text{surface area of sphere with volume } (V)}{\text{surface area of sphere with volume } (V)} = \frac{\text{surface area of the actual object } (A)}{\text{surface area of sphere with volume } (V)} - 1 = \frac{A}{(36\pi V^2)^{1/3}} - 1,$$

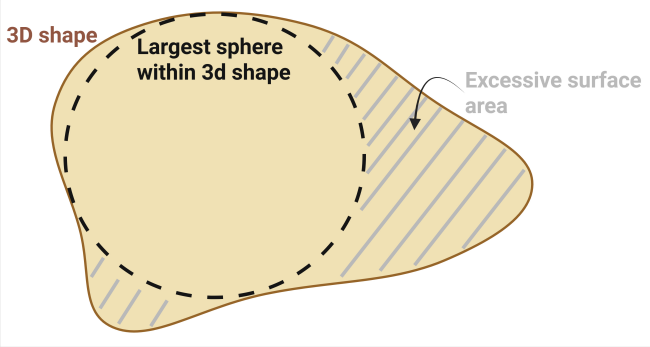
where:

- V = volume of 3D shape in question
- A = surface area of 3D shape in question

▼ **Conceptualization and derivation of mathematical formulation**

▼ **Conceptualization**

RSAE is a ratio of **excessive** surface area of given 3D geometry, relative to that of a sphere having the same volume.



▼ **Derivation of mathematical formulation**

$$RSAE(V, A) = \frac{\text{surface area of the actual object } (A) - \text{surface area of sphere with volume } (V)}{\text{surface area of sphere with volume } (V)}$$

We know that, volume of a sphere with radius r is:

$$V = \frac{4}{3}\pi r^3 \rightarrow (2)$$

We know that surface area of a sphere with radius r is:

$$A = 4\pi r^2 \rightarrow (3)$$

From (2) and (3), the surface area of a sphere with volume V, is:

$$A(V) = 4\pi(V^2) = 4\pi(\frac{4}{3}\pi r^3)^2 = (36\pi V^2)^{1/3} \rightarrow (4)$$

Substituting A(V) from (4) in (1), we get:

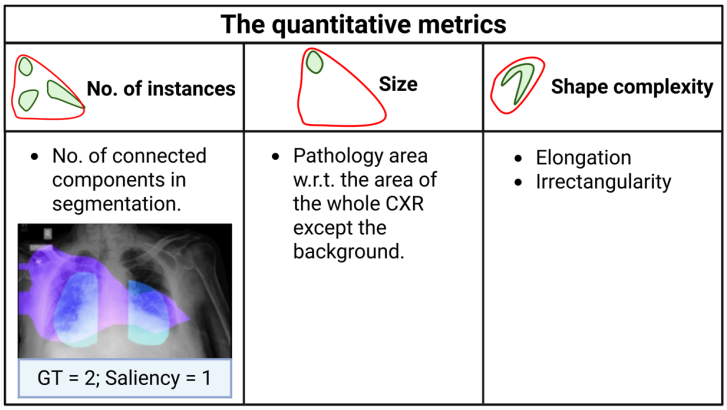
$$RSAE(V, A) = \frac{\text{surface area of the actual object } (A) - \text{surface area of sphere with volume } (V)}{\text{surface area of sphere with volume } (V)} = \frac{\text{surface area of the actual object } (A)}{\text{surface area of sphere with volume } (V)} - 1 = \frac{A}{(36\pi V^2)^{1/3}} - 1$$

▼ **Features of shape-based complexity**

- Honestly, I do not know what features are best here.
- But, I want to follow an active learning-based approach here.

- For the time being, let us just use the following features as placeholders (this idea was inspired by the Saporta et al 2022 paper [7]) — as a placeholder, I list (and explain) their features. **[Please note that the exact features may change based on the segmentation results.]**

    1. No. of connected components
    2. Size of segmentation (**NOT** the same metrics as described under "Shape-agnostic features" — we are interested in 3D statistics here and everywhere else in the analyses, as described above):
        a. Volume
        b. Surface area
        c. Largest radius completely inside 3D geometry
    3. Irrectangularity



▼ **2.2.3 Magnitude and direction of error relative to the organ — further on anisotropy (deformation)**

▼ **Shape preservation metric, Metric 1** (I am proposing this, not an established metric)

[Note: This metric does not have a name. It is a metric that **I am proposing**. Let us just call it **Metric 1**.]

▼ **Motivation behind this metric — *Why?***

I am trying to develop a combined metric of **1. anisotropy** and **2. direction** of the aforementioned anisotropy.

- One metric that is to be used for the comparison between shapes — in our case, the ground truth and the predicted segmentations.
- I wanted to make sure that this metric is size-agnostic — meaning, we can directly compare the **1. magnitude** and **2. direction** of anisotropy of predicted segmentation relative to the organ, **across different sizes of organs without further size-based normalization**.
- It must be noted that this is a **metric of anisotropy, not surface irregularity/ roughness**. In other words, it is a **global metric, not a local metric**. What that means is, the metric must amplify whole-organ shape statistics, and either ignore or suppress local roughness/ irregularity statistics.

  ⇒ Insensitive to local shape statistics, sensitive to global shape statistics.

Metric 1

Quantification of anisotropy (elongation)

Directional alignment of anisotropy (elongation)

Normalized by size

Makes comparison between shapes possible:
- Ground truth and predicted segmentations
- Measure of relative directional anisotropy b/w ground truth and predicted segmentations, comparable across organs of varying sizes.

Insensitive to local statistics (surface roughness)

▼ **Each component separately**

▼ **1. Tensor representation of the 3D geometry**

▼ **Goal**

The tensor represents:

i. Direction of maximum elongation.

ii. How elongated the shape is in each axis (x, y and z).

▼ **Conceptualization and calculation of the shape tensors**

▼ Given, a 3D shape

- Let, each voxel on the 3D shape be represented by:

  $\vec{x_i} = \{x_i, y_i, z_i\}$.

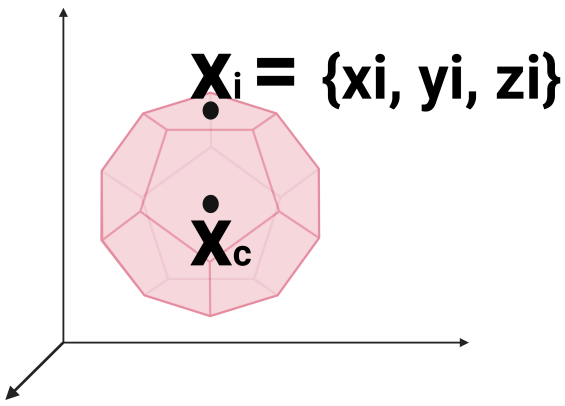
- Let, the centroid be represented by:

  $\vec{x_c} = \{x_c, y_c, z_c\}$.

  The centroid $\vec{x_c}$ is the center of mass (average of all voxels) on the 3D geometry:

  - $x_c = \frac{1}{N}\sum_i x_i$
  - $y_c = \frac{1}{N}\sum_i y_i$
  - $z_c = \frac{1}{N}\sum_i z_i$

  where, N is the total number of voxels on the 3D shape.



- Next, I compute the shape tensor:
  $T = \frac{1}{N}\sum_{i=1}^{N}\{{x_i}^{\rightarrow} - x^{-}\}\{x^{-} - {x_i}^{\rightarrow}\}^T$
  where, $\{{x_i}^{\rightarrow} - x^{-}\}$ is the vector difference between each voxel on the 3D shape and the centroid.
- Let, the difference vector $\{{x_i}^{\rightarrow} - x^{-}\} = \begin{bmatrix} a \\ b \end{bmatrix}$
- Then, $\{{x_i}^{\rightarrow} - x^{-}\}\{x^{-} - {x_i}^{\rightarrow}\}^T = \begin{bmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{bmatrix}$
- Information contained in matrix $\{{x_i}^{\rightarrow} - x^{-}\}\{x^{-} - {x_i}^{\rightarrow}\}^T$:
  - The elongation in each of the three axes x, y and z.
  - The correlation between the $\{x, y\}$, $\{y, z\}$ and $\{x, z\}$ axes.

  $\{{x_i}^{\rightarrow} - x^{-}\}\{x^{-} - {x_i}^{\rightarrow}\}^T =$
  $\begin{bmatrix} elongation(x) & correlation(x,y) & correlation(x,y) \\ correlation(y,x) & elongation(y) & correlation(y,z) \\ correlation(z,x) & correlation(z,y) & elongation(z) \end{bmatrix}$
  ⇒ ***Therefore*** $\{{x_i}^{\rightarrow} - x^{-}\}\{x^{-} - {x_i}^{\rightarrow}\}^T$ ***is the matrix representation of the shape of the organ in terms of the extent of elongation in each of the three dimensions*** $\{x, y, z\}$.

- Finally, I calculate eigenvectors and eigen values associated with the 3D shape.
  - Eigen vector 1 $({v_1}^{\rightarrow})$ provides the direction of the highest elongation of the object (in other word, the length of the object).
    Eigen vectors 2 $({v_2}^{\rightarrow})$ and 3 $({v_2}^{\rightarrow})$ are perpendicular to 1 $({v_1}^{\rightarrow})$. ${v_2}^{\rightarrow}$ and $({v_3}^{\rightarrow})$ provide the direction of highest and second-highest elongation, perpendicular to $({v_1}^{\rightarrow})$.
    ⇒ ***In other words, the eigenvectors basically provide the directions of the length (principal component with highest elongation), breadth and height of a 3D shape***.
  - Eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ give us the magnitude of elongation along the respective directions, which are the eigenvectors $\{{v_1}^{\rightarrow}, {v_2}^{\rightarrow}, {v_3}^{\rightarrow}\}$ defined above.
    ⇒ ***Note that*** $\lambda_1 > \lambda_2 > \lambda_3$.

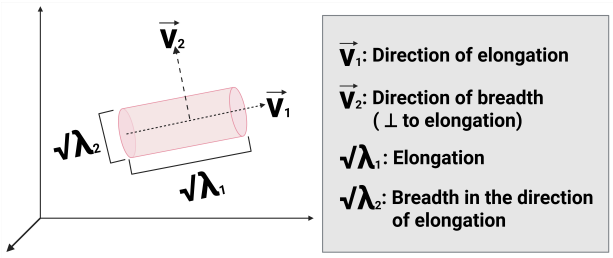
▼ 2. Calculate the difference in shape deformation between the ground truth and predicted segmentations

- **Note 1**: We use the difference in ratio of elongation between the first two eigenvalues $\sqrt{\frac{\lambda_2}{\lambda_1}}$, between the ground truth and the predicted segmentations — as a proxy for shape deformation.
- **Note 2**: We do **not** incorporate **direction** in this step yet. The ratio of $\sqrt{\frac{\lambda_2}{\lambda_1}}$ has been calculated separately for the ground truth and the predicted segmentations.
- **Note 3**: We incorporate direction in the next step.

▼ Goal

The goal is to find the difference between the degree of deformation ($\frac{\text{breadth in the direction of maximum elongation}}{\text{maximum elongation}}$) between the ground truth and the predicted segmentations.

⇒ This gives us the difference in deformation basically: "How much more deformed is the prediction w.r.t. the ground truth?"



▼ Conceptualization and calculation of the **difference in shape deformation between the ground truth and predicted segmentations**

$$\text{Elongation (E)} = \frac{\text{length of second-largest axis}}{\text{length of argest axis}} = \sqrt{\frac{\lambda_2}{\lambda_1}} \rightarrow (1)$$

What we need is, $\text{Difference in elongation}(\Delta E) = E_{\text{pred}} - E_{\text{GT}} \rightarrow (2)$

▼ 3. Find alignment (direction) between the major axes of the GT and the predicted segmentations — "How aligned are they?"

Alignment $(\phi) = \cos^{-1}|v_{1\text{GT}}{}^{\rightarrow}.v_{1\text{pred}}{}^{\rightarrow}|$

▼ Range of ϕ

$\phi \epsilon [0, \frac{\pi}{2}]$

- 0, when $v_1\text{GT}^{\rightarrow}$ and $v_1\text{pred}^{\rightarrow}$ are **aligned (parallel)**.
- $\frac{\pi}{2}$, when $v_1\text{GT}^{\rightarrow}$ and $v_1\text{pred}^{\rightarrow}$ are **unaligned (perpendicular)**.

▼ Summary of the elongation and alignment metrics so far

In order to have a complete shape-based comparison between the GT and the predicted segmentations, we could report:

1. DSC/ NSD/ IoU → for overlap, **and**

2. ΔE → for magnitude of elongation, **and**

3. ϕ → for alignment of elongation, **between the GT and the predicted segmentations**.

⇒ This gives us a pretty complete shape-based metric of comparison. However, we want to combine all of these into one score. See next step.

▼ 4. Bringing it all together

Shape preservation metric, Metric 1:

$M1 = e^{-\Delta E}|v_1\text{GT}^{\rightarrow}.v_1\text{pred}^{\rightarrow}|$

▼ Further explanations on Shape preservation metric, $M1$

▼ Range of $M1$

$M1 \epsilon [0, 1]$:

- 1, when shape is perfectly preserved between the GT and the predicted segmentations
- 0, when shape is completely lost between the GT and the predicted segmentations

▼ **2.2.4 Scalability analyses**

Which organ-specific features are scalable, and by how much?

▼ Scalability of explanatory features from a per-organ basis to across organs

1. [comparative analyses] What are the exceptions?

2. [explanatory/ causal analyses] Why the exceptions?

▼ Analysis framework

Note: The exact strategy and/ or techniques for analysis of this part is quite difficult to guess — as this is the last step. But, I will list out some pointers here. Please note that these are likely to change.

▼ Example framework 1

Example data:

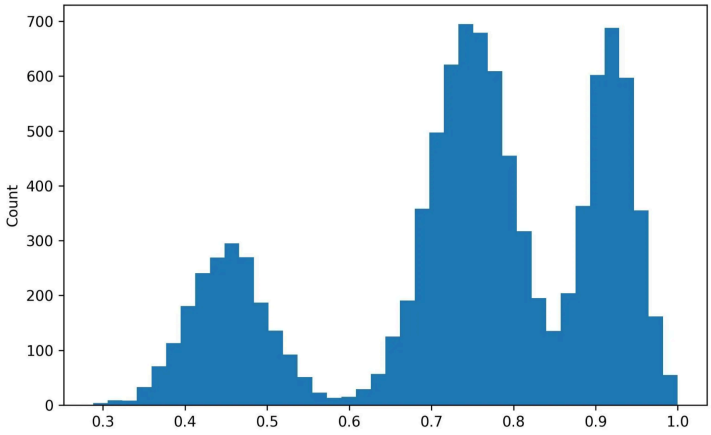


Fig. Histogram of error score (β) values

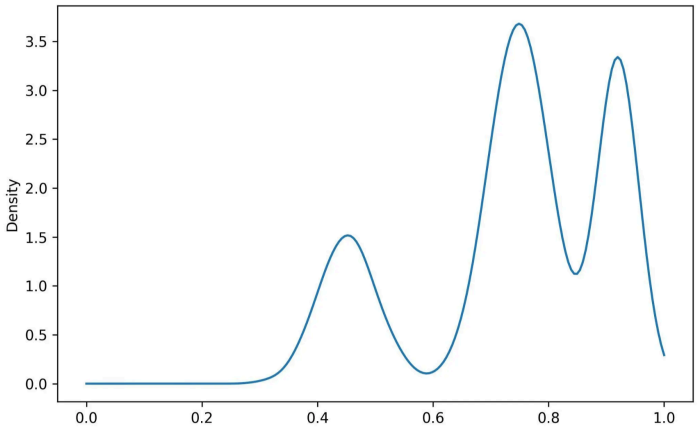


Fig. KDE of error score (β) values

Step 1: Fit a Gaussian mixture model (GMM) to the error score (β) per organ. Subsequently, find optimal number of components (N) in the GMM-fitted data, as guided by BIC.

In this example:

> The lowest BIC is with 3 components.

- GMM cluster means:
  - Low β: 0.45
  - Medium β: 0.75
  - High β: 0.92
- Cluster sizes:
  - Low β: 1998 patients
  - Medium β: 4945 patients
  - High β: 3057 patients

Step 2: Extract shape- and texture-based features from each of the three components corresponding to "under," "mixed" and "over" segmentation — based on error score (β)-DSC combinations.

Step 3: Perform feature selection to find features that are most important in classifying "under," "mixed" and "bad" segmentations.

⇒ Example: Say, livers that have low circularity, high area and high circumference are associated with "over" segmentations.

2.3 Going one level deeper, for feature values that have been identified as causing errors in under/ mixed/ over segmentation, can we tell if they lead to a specific type of under/ mixed/ over segmentation?

⇒ <Example> From 5.3, step 3: Say, livers that have low circularity, high area and medium circumference are associated with "over" segmentations.

Step 1: Label each "over" segmentation as "low deformation" (preserves GT shape — low ΔE), or "high deformation" (does not preserve GT shape — high ΔE).

Step 2. Fit a decision tree classifier to identify what combinations of feature values lead to low versus high deformation.

⇒ This could be done for any number of aforementioned features — often in the order that we are interested in to get more fine-grained insights.

→ Of course, I would need to look at some initial results to see if this method actually works.

→ We could use a decision tree-based method (for instance, random forest — which is a boosted decision tree algorithm — for this purpose).

▼ 3. Literature review

▼ 4.1 Tables

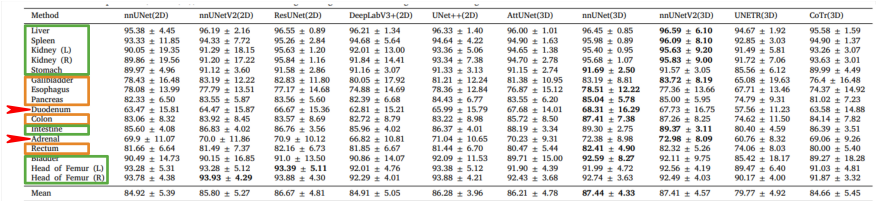
	Dice similarity coefficients (mean ± standard deviation)								
	Training test set			External test set—DECT			External test set—SECT		
	DE-PVP	VNC	P value*	DE-PVP	VNC	P value*	SE-PVP	TNC	P value*
Liver	0.976 ± 0.004	0.973 ± 0.007	0.206	0.986 ± 0.003	0.977 ± 0.006	< 0.001	0.981 ± 0.005	0.965 ± 0.012	< 0.001
Spleen	0.963 ± 0.012	0.959 ± 0.017	0.280	0.978 ± 0.011	0.966 ± 0.016	< 0.001	0.972 ± 0.015	0.961 ± 0.019	< 0.001
Right kidney	0.954 ± 0.017	0.945 ± 0.018	< 0.001	0.976 ± 0.006	0.968 ± 0.008	< 0.001	0.971 ± 0.007	0.944 ± 0.016	< 0.001
Left kidney	0.955 ± 0.015	0.950 ± 0.018	0.135	0.976 ± 0.007	0.970 ± 0.008	< 0.001	0.970 ± 0.007	0.954 ± 0.011	< 0.001
Pancreas	0.879 ± 0.045	0.849 ± 0.046	0.001	0.873 ± 0.045	0.854 ± 0.046	< 0.001	0.846 ± 0.060	0.810 ± 0.045	< 0.001

DECT dual-energy CT, *SECT* single-energy CT, *DE-PVP* portal venous phase on dual-energy CT, *VNC* virtual non-contrast, *SE-PVP* portal venous phase on single-energy CT, *TNC* true non-contrast. *P-values were calculated using a paired *t*-test (DE-PVP vs. VNC and SE-PVP vs. TNC).

Table L1 — from [1]. Dice similarity coefficients of the 3D nnU-Net-based algorithm in abdominal organ segmentation.

	Intraclass correlation coefficient (95% confidence interval)					
	Training test set		External test set—DECT		External test set—SECT	
	DE-PVP	VNC	DE-PVP	VNC	SE-PVP	TNC
Liver	0.999 (0.995, 0.999)	0.999 (0.997, 0.999)	0.999 (0.998, 0.999)	0.999 (0.997, 0.999)	0.998 (0.996, 0.999)	0.985 (0.772, 0.996)
Spleen	0.999 (0.996, 0.999)	0.999 (0.996, 0.999)	0.999 (0.998, 0.999)	0.999 (0.997, 0.999)	0.999 (0.998, 0.999)	0.992 (0.984, 0.996)
Right kidney	0.925 (0.697, 0.981)	0.916 (0.660, 0.979)	0.991 (0.957, 0.997)	0.992 (0.984, 0.996)	0.989 (0.961, 0.996)	0.971 (0.816, 0.991)
Left kidney	0.927 (0.704, 0.982)	0.914 (0.653, 0.979)	0.983 (0.884, 0.995)	0.981 (0.935, 0.993)	0.982 (0.839, 0.994)	0.970 (0.454, 0.993)
Pancreas	0.892 (0.566, 0.973)	0.887 (0.546, 0.972)	0.889 (0.689, 0.954)	0.890 (0.395, 0.965)	0.792 (0.231, 0.926)	0.840 (0.694, 0.920)

Table L2 — from [1]. Agreement — intraclass correlation coefficient (ICC) — between ground truth organ segmentation and that by 3D nnU-Net-based algorithm for liver, spleen, kidneys and pancreas.



Method	nnUNet(2D)	nnUNetV2(2D)	ResUNet(2D)	DeepLabV3+(2D)	UNet++(2D)	AttUNet(3D)	nnUNet(3D)	nnUNetV2(3D)	UNETR(3D)	CoTr(3D)
Liver	95.38 ± 4.45	96.19 ± 2.16	96.55 ± 0.89	96.21 ± 1.34	96.33 ± 1.40	96.00 ± 1.01	96.45 ± 0.85	**96.59 ± 6.10**	94.67 ± 1.92	95.58 ± 1.59
Spleen	93.33 ± 11.85	94.33 ± 7.72	95.26 ± 2.84	94.68 ± 5.64	94.64 ± 4.22	94.90 ± 1.63	95.98 ± 0.89	**96.09 ± 8.10**	92.85 ± 3.03	94.90 ± 1.37
Kidney (L)	90.05 ± 19.35	91.29 ± 18.15	95.63 ± 1.20	92.01 ± 13.00	93.36 ± 5.06	94.65 ± 1.38	95.40 ± 0.95	**95.63 ± 9.20**	91.49 ± 5.81	93.26 ± 3.07
Kidney (R)	89.86 ± 19.56	91.20 ± 17.22	95.84 ± 1.16	91.84 ± 14.41	93.34 ± 7.38	94.70 ± 2.78	95.68 ± 1.07	**95.83 ± 9.00**	91.72 ± 7.06	93.63 ± 3.01
Stomach	89.97 ± 4.96	91.12 ± 3.60	91.58 ± 2.86	91.16 ± 3.07	91.33 ± 3.13	91.15 ± 2.74	**91.69 ± 2.50**	91.57 ± 3.05	85.56 ± 6.12	89.99 ± 4.49
Gallbladder	78.43 ± 16.48	83.19 ± 12.22	82.83 ± 11.80	80.05 ± 17.92	81.21 ± 12.24	81.38 ± 10.95	83.19 ± 8.81	**83.72 ± 8.19**	65.08 ± 19.63	76.4 ± 16.48
Esophagus	78.08 ± 13.99	77.79 ± 13.51	77.17 ± 14.68	74.88 ± 14.69	78.36 ± 12.84	76.87 ± 15.12	**78.51 ± 12.22**	77.36 ± 13.66	67.71 ± 13.46	74.37 ± 14.92
Pancreas	82.33 ± 6.50	83.55 ± 5.87	83.56 ± 5.60	82.39 ± 6.68	84.43 ± 6.77	83.55 ± 6.20	**85.04 ± 5.78**	85.00 ± 5.95	74.79 ± 9.31	81.02 ± 7.23
Duodenum	63.47 ± 15.81	64.47 ± 15.87	66.67 ± 15.36	62.81 ± 15.21	65.99 ± 15.79	67.68 ± 14.01	**68.31 ± 16.29**	67.73 ± 16.75	57.56 ± 11.23	63.58 ± 14.88
Colon	83.06 ± 8.32	83.92 ± 8.45	83.57 ± 8.69	82.72 ± 8.79	83.22 ± 8.98	85.72 ± 8.50	**87.41 ± 7.38**	87.26 ± 8.25	74.62 ± 11.50	84.14 ± 7.82
Intestine	85.60 ± 4.08	86.83 ± 4.02	86.76 ± 3.56	85.96 ± 4.02	86.37 ± 4.01	88.19 ± 3.34	89.30 ± 2.75	**89.37 ± 3.11**	80.40 ± 4.59	86.39 ± 3.51
Adrenal	69.9 ± 11.07	70.0 ± 11.86	70.9 ± 10.12	66.82 ± 10.81	71.04 ± 10.65	70.23 ± 9.31	72.38 ± 8.98	**72.98 ± 8.09**	60.76 ± 8.32	69.06 ± 9.26
Rectum	81.66 ± 6.64	81.49 ± 7.37	82.16 ± 6.73	81.85 ± 6.67	81.44 ± 6.70	80.47 ± 5.44	**82.41 ± 4.90**	82.32 ± 5.26	74.06 ± 8.03	80.00 ± 5.40
Bladder	90.49 ± 14.73	90.15 ± 16.85	91.0 ± 13.50	90.86 ± 14.07	92.09 ± 11.53	89.71 ± 15.00	**92.59 ± 8.27**	92.11 ± 9.75	85.42 ± 18.17	89.27 ± 18.28
Head of Femur (L)	93.28 ± 5.31	93.28 ± 5.12	**93.39 ± 5.11**	92.01 ± 4.76	93.38 ± 5.12	91.90 ± 4.39	91.99 ± 4.72	92.56 ± 4.19	89.47 ± 6.40	91.03 ± 4.81
Head of Femur (R)	93.78 ± 4.38	**93.93 ± 4.29**	93.88 ± 4.30	92.29 ± 4.01	93.88 ± 4.21	92.43 ± 3.68	92.74 ± 3.63	92.49 ± 4.03	90.17 ± 4.00	91.87 ± 3.32
Mean	84.92 ± 5.39	85.80 ± 5.27	86.67 ± 4.81	84.91 ± 5.05	86.28 ± 3.96	86.21 ± 4.78	**87.44 ± 4.33**	87.41 ± 4.57	79.77 ± 4.92	84.66 ± 5.45

Table L3 — from [2]. DSCs of 16 abdominal organs segmented using 10 popular segmentation methods.

Organs	Liver	Spleen	Kidney (L)	Kidney (R)	Stomach	Gallbladder	Esophagus	Pancreas
Senior	1.73 ± 0.03	1.34 ± 0.12	1.28 ± 0.09	1.36 ± 0.11	3.95 ± 0.52	6.86 ± 1.29	7.65 ± 2.24	7.49 ± 2.39
Expert	0.89 ± 0.01	0.93 ± 0.03	1.03 ± 0.08	0.97 ± 0.07	1.48 ± 0.35	2.97 ± 0.94	3.32 ± 1.25	2.89 ± 0.91
Organs	Duodenum	Colon	Intestine	Adrenal	Rectum	Bladder	Head of Femur (L)	Head of Femur (R)
Senior	9.67 ± 4.11	4.74 ± 2.35	3.66 ± 1.26	9.86 ± 4.38	3.76 ± 1.44	2.73 ± 0.89	1.78 ± 0.94	1.63 ± 0.46
Expert	3.33 ± 1.23	1.27 ± 0.23	1.48 ± 0.69	3.75 ± 1.73	1.46 ± 0.74	1.35 ± 0.39	0.96 ± 0.13	0.87 ± 0.09

Table L4 — from [2]. Mean difference in DSC between a senior radiologist (7+ years of experience) and an expert radiologist (20+ years of experience), and the consensus segmentation.

Organ	**DSC**
Spleen	0.983
Aorta	0.981
Liver	0.965
Left kidney	0.953
Stomach	0.947
Esophagus	0.944
Right kidney	0.939
Bladder	0.934
Right adrenal gland	0.909
Left adrenal gland	0.898
Colon	0.896
Pancreas	0.887
Gallbladder	0.875

Table L5 — from TotalSegmentator [4]. DSC on abdominal organs of interest (see ***Section 1. Introduction > Organs that we are focusing on***).

Organ	NSD
Aorta	0.997
Right adrenal gland	0.992
Left adrenal gland	0.985
Spleen	0.985
Esophagus	0.974
Liver	0.973
Left kidney	0.962
Gallbladder	0.958
Pancreas	0.958
Right kidney	0.956
Stomach	0.946
Bladder	0.944
Colon	0.921

Table L6 — from TotalSegmentator [4]. NSD on abdominal organs of interest (***Section 1. Introduction > Organs that we are focusing on***).

▼ 4.2 Figures

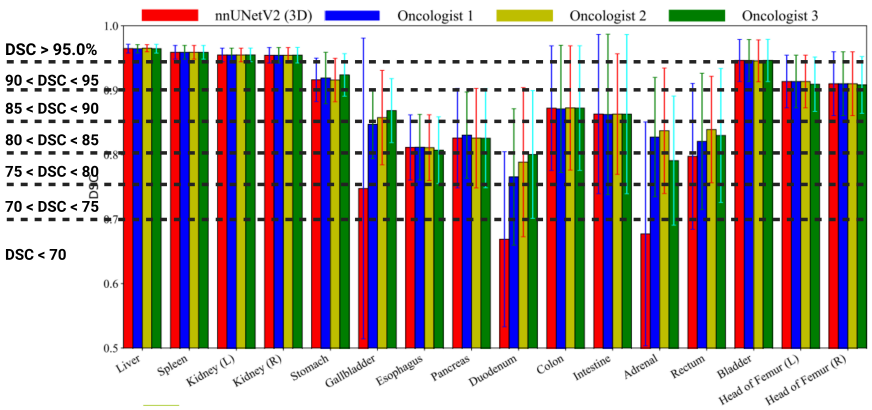


Fig L1 — from [2]. DSC by nnUNetV2, and DSC improved by 3 junior radiologists (3 years of experience).

▼ **4. Notes for Surya**

[**Note**: This section is meant for me (Surya), for the purpose of keeping track of the various reasons and rationales behind some of the decisions we made, so we can explain this better if asked by reviewers.]

▼ Expand ***Section 4. Notes for Surya***

4.1 Why were these five organs selected?

▼ Expand ***Section 4.1***

We wanted to select two easy (liver, kidneys) and three difficult to segment organs (pancreas, esophagus, inferior vena cava) — to ensure that our method was robust, and not only suitable for one type of geometry.

▼ **5. Version control**

▼ Expand ***Section 5. Version control***

▼ Version 1

error-characterization-protocol (2).pptx

▼ Version 2

error-characterization-protocol.pptx

▼ **References**

1. Jeon, Sun Kyung, Ijin Joo, Junghoan Park, Jong-Min Kim, Sang Joon Park, and Soon Ho Yoon. 2024. “Fully-Automated Multi-Organ Segmentation Tool Applicable to Both Non-Contrast and Post-Contrast Abdominal CT: Deep Learning Algorithm Developed Using Dual-Energy CT Images.” *Scientific Reports* 14 (1): 4378.
2. Luo, Xiangde, Wenjun Liao, Jianghong Xiao, Jieneng Chen, Tao Song, Xiaofan Zhang, Kang Li, Dimitris N. Metaxas, Guotai Wang, and Shaoting Zhang. 2022. “WORD: A Large Scale Dataset, Benchmark and Clinical Applicable Study for Abdominal Organ Segmentation from CT Image.” *Medical Image Analysis* 82 (102642): 102642.
3. Ji, Yuanfeng, Haotian Bai, Jie Yang, Chongjian Ge, Ye Zhu, Ruimao Zhang, Zhen Li, et al. 2022. “AMOS: A Large-Scale Abdominal Multi-Organ Benchmark for Versatile Medical Image Segmentation.” *arXiv [Eess.IV]*. arXiv. https://proceedings.neurips.cc/paper_files/paper/2022/file/ee604e1bedbd069d9fc9328b7b9584be-Paper-Datasets_and_Benchmarks.pdf.
4. Wasserthal, Jakob, Hanns-Christian Breit, Manfred T. Meyer, Maurice Pradella, Daniel Hinck, Alexander W. Sauter, Tobias Heye, et al. 2023. “TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images.” *Radiology. Artificial Intelligence* 5 (5): e230024.
5. Bionetworks, Sage. n.d. “Multi-Atlas Labeling Beyond the Cranial Vault - Workshop and Challenge.” Accessed December 16, 2025. https://www.synapse.org/Synapse:syn3193805/wiki/89480.
6. “MICCAI FLARE 2022 - Grand Challenge.” n.d. Grand-challenge.org. Accessed December 16, 2025. https://flare22.grand-challenge.org/.
7. Saporta, Adriel, Xiaotong Gui, Ashwin Agrawal, Anuj Pareek, Steven Q. H. Truong, Chanh D. T. Nguyen, Van-Doan Ngo, et al. 2022. “Benchmarking Saliency Methods for Chest X-Ray Interpretation.” Nature Machine Intelligence 4 (10): 867–78.

▼ **[SIIM abstract] Categorizing deep learning-based 3D segmentation errors across organs**

1. Purpose

There is a dearth of literature that systematically benchmarks organs for the kinds of errors that AI-based segmentation models are likely to make while performing segmentation on CT and/ or MRI scans. Consequently, we lack understanding of image features that lead to specific kinds of AI segmentation errors. Metrics that define segmentation quality such as Dice similarity coefficient (DSC) or Intersection over union (IoU), do not tell us what kind of error the model is making — only the extent of the error made. The purpose of this work is to devise a quantitative metric that indicates the degree of over- and under-segmentation error made by AI models, and subsequently identify organ-specific features that are susceptible to such errors.

2. Materials and methods

In this study, we first perform segmentation of five organs (liver, kidney, esophagus, pancreas and inferior vena cava) from four abdominal CT segmentation databases [1-3], using the pre-trained TotalSegmentator model [4]. Next, we formulate a deterministic and completely quantitative metric for measuring the extent of over- and under-segmentation. We call this metric error score, β, which is defined as the difference between the area of the predicted segmentation and the ground truth segmentation, normalized by the area of the ground truth segmentation: $\beta = \frac{Area(predicted) - Area(actual)}{Area(actual)}$. A positive value of β is indicative of over-segmentation, whereas a negative value of β is indicative of under-segmentation. Next, we extract shape-based features from the ground truth segmentation. Subsequently, we perform correlation between the shape-based features and the segmentation score, for each of the five organs separately — in order to elucidate specific shape-based characteristics that make organs susceptible to certain kinds of errors. Specifically, this is done by fitting a simple linear regression model, with each of the features as separate independent variables, and the error score β as the dependent variable. The regression coefficient of the

fitted simple linear regression would tell us how each changing feature affects the extent of over- or under-segmentation. Figure 1 provides the reader with a pictorial overview of the workflow.

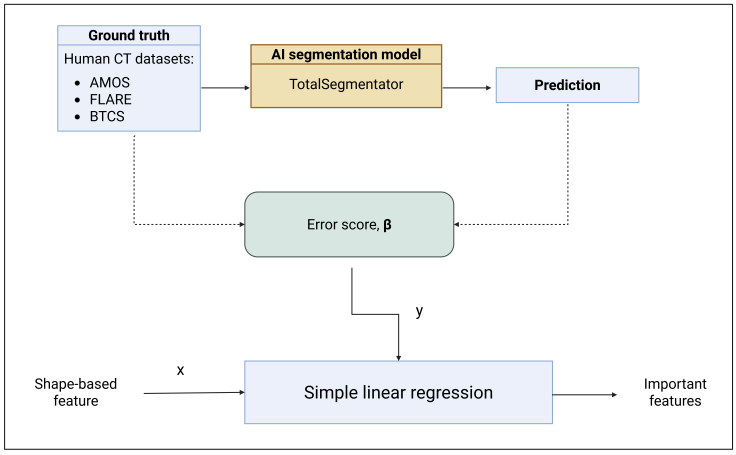


Fig 1. A pictorial overview of the error characterization workflow.

References

1. Ji *et al.*, *AMOS: A Large-Scale Abdominal Multi-Organ Benchmark for Versatile Medical Image Segmentation*, in *Advances in Neural Information Processing Systems 35*, 2022.

2. Bionetworks, Sage. n.d. "Multi-Atlas Labeling Beyond the Cranial Vault - Workshop and Challenge." Accessed December 16, 2025. https://www.synapse.org/Synapse:syn3193805/wiki/89480.

3. "MICCAI FLARE 2022 - Grand Challenge." n.d. Grand-challenge.org. Accessed December 16, 2025. https://flare22.grand-challenge.org/.

4. Wasserthal, Jakob, Hanns-Christian Breit, Manfred T. Meyer, Maurice Pradella, Daniel Hinck, Alexander W. Sauter, Tobias Heye, et al. 2023. "TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images." Radiology. Artificial Intelligence 5 (5): e230024.

▼ **[Error generation protocol and IsBART - deprecated] Error generation protocol for IsBART**

1. Abstract

▼ Expand ***Section 1. Abstract***

Segmentation is used as a pre-processing step for many different medical imaging-based applications. The accuracy of such segmentations have an effect on downstream analyses and results. Small segmentation errors in individual samples have shown to have a significant cumulative impact on genome-wide association studies.

Systematic errors have an especially strong negative impact on genome-wide association studies. Systematic errors are those that result from predictions that consistently either underestimate or overestimate a certain measurement — such that the predicted value has a component of additive bias on the actual value. Systematic errors often mask as genetic traits in GWAS, and are generally much more difficult to identify than random errors. In this project, we try to systematically benchmark IsBART, across different organs, segmentation errors, and databases.

2. A recap of “Location-smoothed Bayesian additive regression trees (lsBART)” by Wooten et al.

▼ Expand ***Section 2. Recap of lsBART***

2.1 Abstract

▼ Expand ***Section 2.1 Abstract***

Accurate segmentation of organs from medical images is an essential pre-processing step used in radiotherapy planning. Automated organ segmantation tools — including bt not restricted to AI-based segmentation models — often make insignificant segmentation errors. These errors, while insignificant on individual patients, can be very misleading when conducting genome-wide association studies (GWAS) that depend on such segmentations as pre-processing steps. Not only are most state-of-the-art AI models susceptble to making such errors, they are also not interpretable. Specifically, they are unable to provide an explanation as to why certain kinds of systematic segmentation errors are more likely to occur than others. Systematic errors are especially devastating to GWAS because they mask as genetic characteristics. Apart from systemtic errors — and the non-interpretability associated with them — lack of robustness is a serious concern. Lack of robustness can either mean the same organ from the exact same image having different segmentation results on different executions of the exact same model, or the same organ having considerably different segmentation results on two different images. lsBART performs quality assurance of automated organ segmentation by deep learning-based models, by scoring the aforementioned models with regards to interpretability and robustness.

2.2 Introduction

▼ Expand ***Expand Section 2.2 Introduction***

In order to score automated organ segmentation models on interpretability and robustness, lsBART tackles the problem of quality assurance as scalar-on-function-regression — where the geometric phenotypes of the ground-truth segmentations are functional predictors, and the response is either “good” or “bad” quality of segmentation predicted by the segmentation model under question.

With the shape-based characteristics as functional predictors and the quality labels of segmentations as responses, lsBART aims to identify: (1) region-specific likelihood of segmentation error (meaning, “where” in a functional predictor the model is likely to make an error); and (2) function-specific likelihood of segmentation error (meaning, given multiple shape-based functions, “which” functions are most responsible for driving such errors).

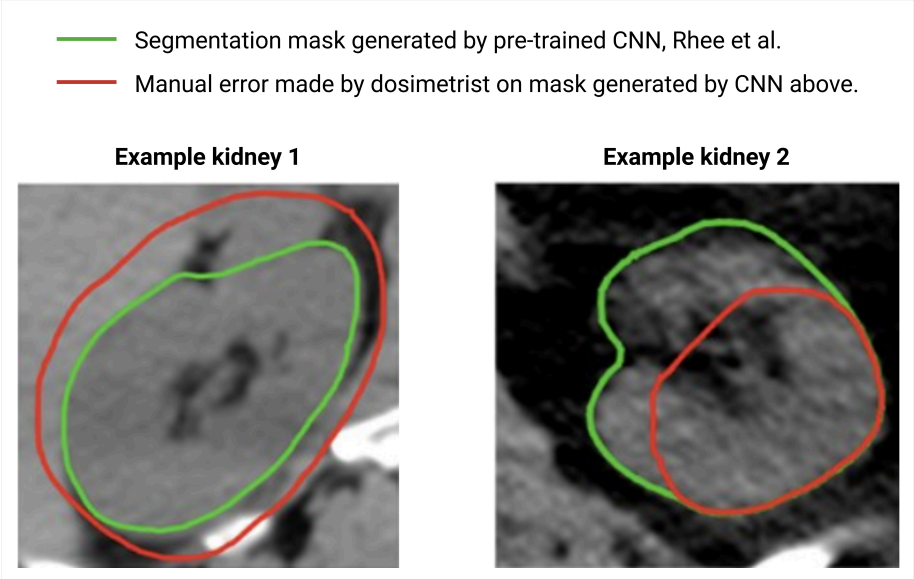
3. Shortcomings of the lsBART model — and how we fill those gaps:

▼ Expand ***Section 3. Shortcomings of lsBART***

3.1 lsBART was not trained to be able to classify true organ segmentation versus AI model-based errors.

▼ Expand ***Section 3.1***

- `lsBART-original` : The lsBART model in the original paper was trained to perform classification between a CNN model-based segmentation (Rhee et al.) masks versus manually generated errors on those masks.



- IsBART-benchmarking:
  - **Noise generation**: We have designed a noise generation algorithm that will generate 6 types of noisy masks:
    1. 3 common types of segmentation errors made by AI segmentation models in general, nnU-Net-based models in particular:
       a. Over-segmentation (unidirectional)
       b. Under-segmentation (unidirectional)
       c. Mixed over- + under-segmentation (unidirectional)
    2. 3 shape-based noisy masks:
       a. Smoothened and/ or shifted
       b. Expansion/ shrinkage with shape preservation
       c. Generation of internal holes or cavities
  - **IsBART-based classification**: We will train IsBART to classify between TotalSegmentator organ segmentation masks versus errors generated on those masks by our noise generation model which is meant to mimic AI model-based errors + shape-based noise. ***(See literature for common AI model-based segmentation errors, Section 4.* Common AI model-based segmentation errors — a literature review.*)***

3.2 IsBART was only trained on one organ — that too, an easy organ to segment by AI-based models, the kidneys.

▼ Expand ***Section 3.2***

- IsBART-original: The IsBART model in the original paper was only trained on one type of organ — that too, an easy organ to segment by AI-based models, the kidneys.
- IsBART-benchmarking: We are training the IsBART model on five different organs:
  - 3 difficult organs to segment by AI-based models on CTs and MRIs

- 2 easy organs to segment by AI-based models on CTs and MRIs

3.2.1 The goal is to be able to answer the following questions:

- Does IsBART perform equally well on easy versus difficult to segment organs?
- For organs that are difficult to segment (meaning, likely to contain more errors by AI-based models as compared to easier organs), can IsBART:
  - Pinpoint the exact areas of error
  - Determine the functional predictors (i.e., the contour shape-based features) that are likely to be responsible for errors across organs.

3.3 IsBART was trained on a very small and very imbalanced dataset

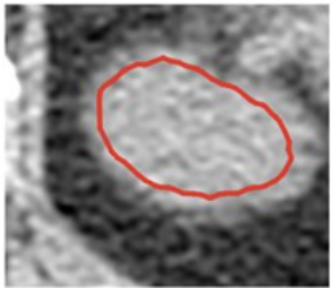
▼ Expand ***Section Section 3.3***

- IsBART-original : 312 kidney contours:—
  - 260 labeled "acceptable" (segmented by pre-trained CNN model, Rhee et al.)
  - 52 labeled "unacceptable" (manually introduced by a dosimetrist)
- IsBART-benchmarking : We will train it on thousands of organs — TotalSegmentator dataset

3.4 Decision of whether an organ segmentation is "acceptable" or "unacceptable", is very subjective.

▼ Expand ***Section 3.4***

- IsBART-original : In the prediction only task (unlabeled, external dataset - MD Anderson):—
  - 18 radiation treatment plans — 36 kidney contours using pre-trained CNN model (Rhee et al.)
  - Example segmentation by Rhee et al:



⇒ Whether or not the segmentation is correct is determined by a dosimetrist.

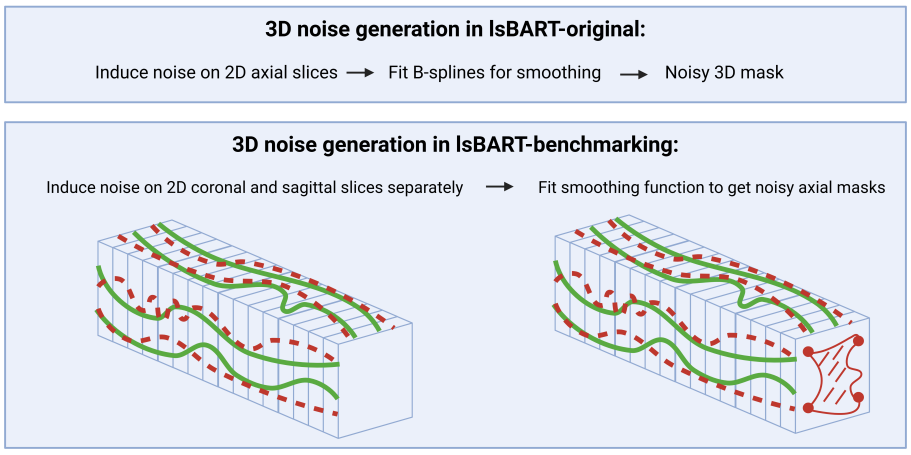
- IsBART-benchmarking : We have TotalSegmentator masks labeled "correct segmentation", and noise generated from these masks labeled "noisy segmentation".

3.5 IsBART does not make 3D errors in the conventional sense — all 3D errors are just "smoothed" interpolations of 2D errors.

▼ Expand ***Section 3.5***

- IsBART-original : inducing error on 2D axial slices, subsequently fit B-splines along the coronal and sagittal planes to have 3D segmentation error.
  - **Disadvantage — Unrealistic and jagged 3D segmentation errors**.

- `IsBART-benchmarking`: I intend to use my "noise generator" algorithm to indice 2D noise on the sagittal and coronal planes — then reconstruct the axial noisy masks.
    - **Advantage — Error more closely resembles state-of-the-art 3D segmentation model-based errors**: This more closely resembles state-of-the-art 3D AI-based segmentation errors — rather than inducing error on 2D axial slices, subsequently fit B-splines along the coronal and sagittal planes to have 3D segmentation error.



3.6 IsBART only takes into account contour shape-based features — no textural features.

▼ Expand ***Section 3.6***

- We also incorporate **texture-based features**, in addition to the original 15 contour shape-based features extracted in the original paper.

4. Common AI model-based segmentation errors — a literature review [Outdated - deprecated]

▼ Expand ***Section 4. Common AI errors***

4.1 Over-segmentation errors (also known as "mask leakage") in AI-based segmentation models from medical images

▼ Expand ***Section 4.1***

- Over-segmentation errors are among the most common types of AI model-based segmentation errors, affecting a wide range of organs and modalities.
- Some organs are more susceptible to over-segmentation (mask leakage) as compared to others — liver is one of those organs that is very susceptible to over-segmentation. The authors of [1] exhibit that over-segmentation of the liver often leads to a higher Dice score than under-segmentation, for the same refernce ground truth. Therefore, liver segmentation models trained to maximize Dice scores, tend to over-segment. This has real-life implications — namely, over-segmentation of the liver by AI-based segmentation models, are often respondible for higher chances of resection and/ or ablation of healthy tissue during surgeries [1].
- Small arteries and tumors of the kidneys are often over-segmented by state-of-the-art renal segmentations models such as RenalSegNet [2].
- The authors of [3] have exhibited that class imbalance often causes over-segmentation of the lungs.

- Additional literature pertinent with over-segmentation errors made by AI-based models on medical images: [4].

4.1.1 Over-segmentation errors (also known as “mask leakage”) — specifically in nnU-Net-based models

▼ Expand ***Section 4.1.1***

- In whole-body PET/CT segmentation tasks using nnU-Net, several papers report over-segmentation. For example, in the “AutoPET Challenge 2023: nnU-Net-based whole-body 3D PET-CT Tumour Segmentation” paper [5], the authors state that automated segmentation methods often struggle with over-segmentation of regions of healthy metabolic activity.” They go on to report a signififcant false positive volume (about 5.78 mL) on their internal test set.
- The authors of [6] exhibit that in MRI segmentation of ocular adnexal lymphoma using nnU-Net, over-segmentation is a pervasive issue.

4.2 Under-segmentation errors in AI-based segmentation models from medical images

▼ Expand ***Section 4.2***

- Much like over-segmentation, under-segmentation is a pervasive issue with AI-based models — spanning many different organs and modalities.
- The authors of [7] have exhibited that their pancreas segmentation model is likely to under-segment by omiting pixels from the boundaries of the pancreas.
- The authors of [8] have exhibited that under-segmentation is more so an issue with smaller organs as compared to larger ones. The authors of [9] concur with such findings — also exhibiting that smaller organs are morelikely to be under-segmented than bigger ones. Additionally, they also show that class imbalance leads to both over- and under-segmentation.

4.2.1 Under-segmentation errors in nn-U-Net based segmentation models specifically

▼ Expand ***Section 4.2.1***

- Under-segmentation in automatic segmentation of fetal brain tissue [10]
- Under-segmentation in automatic segmentation of adult brain tissue [11]

4.3 Feedback

▼ Expand ***Section 4.3***

- Literature does not include quality papers
- Also, based on domain knowledge and clinical experience, Paul (and Pritam) know that there is a dearth of systematic error characterization.
- Therefore, instead of trying to replicate thier “types of errors,” we should just go with over- and under-segmentation first.

5. Noise generation

▼ Expand ***Section 5. Noise generation***

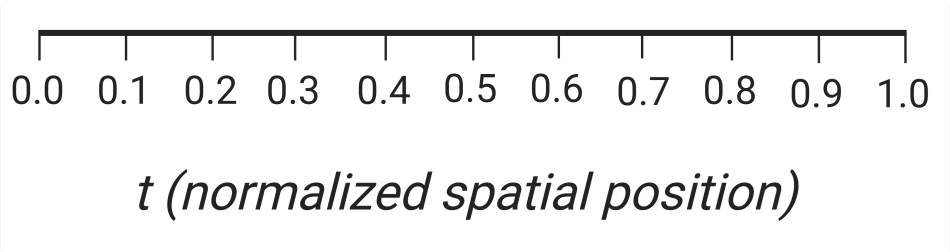
5.1 Noise generation in original paper

▼ Expand ***Section 5.1 Noise generation in original paper***

5.1.1 Simulation study — noise generation on functional predictor

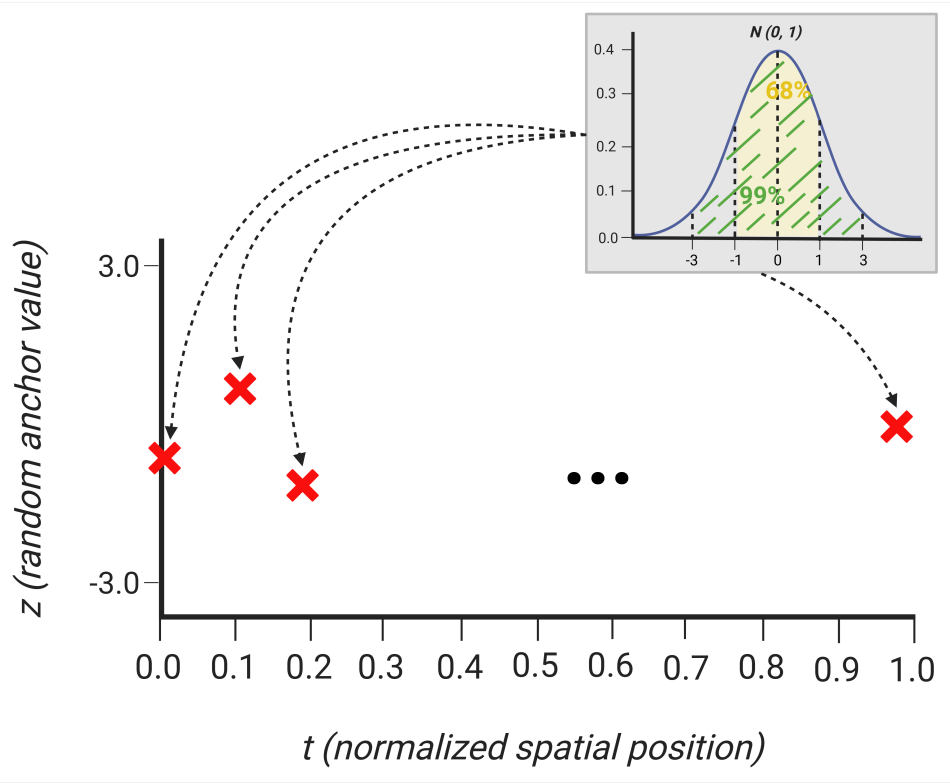
▼ **5.1.1.1 Data and noise generation in simulation study**

▼ **Step 1 — choosing anchor location**: Choose 10 equally spaced grid points from the interval $t \in [0, 1]$. These 10 points will serve as anchor locations $t = \{t_1, t_2, \ldots, t_{10}\}$.



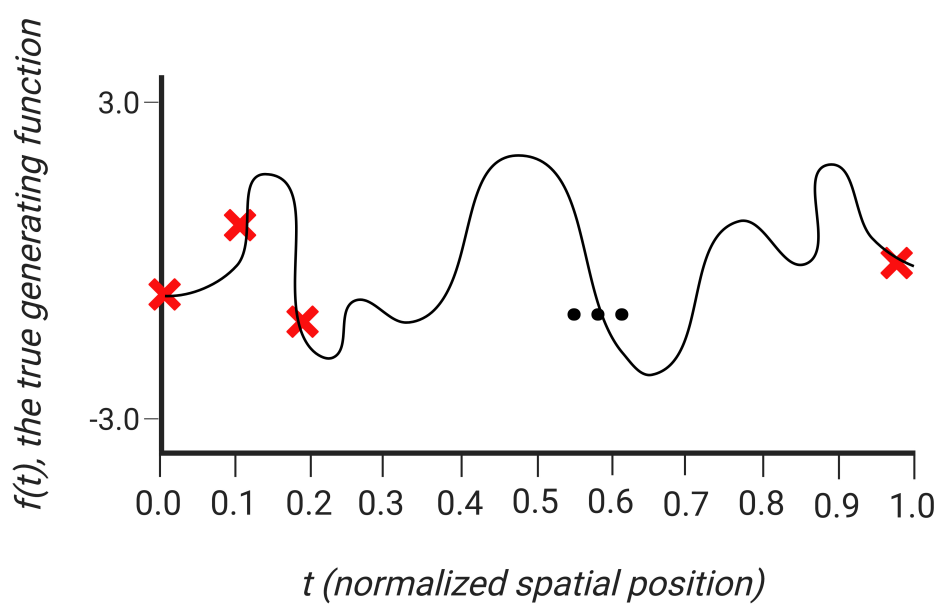
▼ **Step 2 — generating random anchor values**: For each t_j, select a random value $z_j \sim \mathcal{N}(0, 1)$.

(t_j)	0.0	0.1	0.2	...	1.0
(z_j)	0.3	−0.7	1.4	...	−0.1



▼ **Step 3 — fitting a smooth function through the anchor values**: Fit the anchor points with B-splines using 8 basis functions — thereby interpolating between them:

$$f(t) = \sum_{k=1}^{8} \beta_k B_k(t), \text{ where } \beta_k \text{ are fitted such that} f(t_j) \approx z_j.$$



▼ **Step 4 — generating each individual functional predictor** $X_i(t)$: $f(t)$ is a smooth generating function that represents a functional predictor (shape-based feature) extracted from an organ segmentation in a medical image. $f(t)$ acts as a template for all simulated functional predictors $X = X_i$:

$$X_i(t) = f(t) + \varepsilon_i(t),$$

where $\varepsilon_i(t) \sim N(0, \sigma^2)$ independently at each of 100 evaluation points along $[0, 1]$. It must be noted that for each true simulation function $f(t)$, a set of 10 noisy signals $f(t) + \varepsilon_i(t)$ are generated.

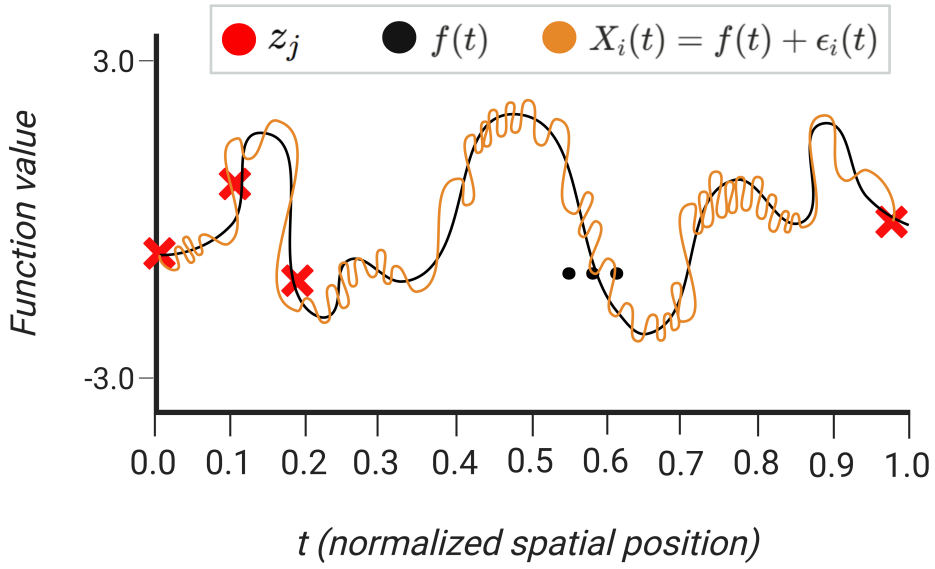
- Generation of $\varepsilon_i(t)$:
  - Divide the interval $[0, 1]$ into 100 equally spaced points $t_1, t_2, \ldots, t_{100}$.
  - The random variables at each point, $\varepsilon_i(t_1), \varepsilon_i(t_2), \ldots, \varepsilon_i(t_{100})$, are **independent draws** from the same $N(0, \sigma^2)$ distribution.
  - There is **no correlation** between the errors at different points along t.
- Python code for generation of $\varepsilon_i(t)$:

```
import numpy as np

sigma = 1.0
n_points = 100

epsilon_i = np.random.normal(0, sigma, n_points)
```

[Note: In this code block, `epsilon_i[j]` is $\varepsilon_i(t_j)$.]

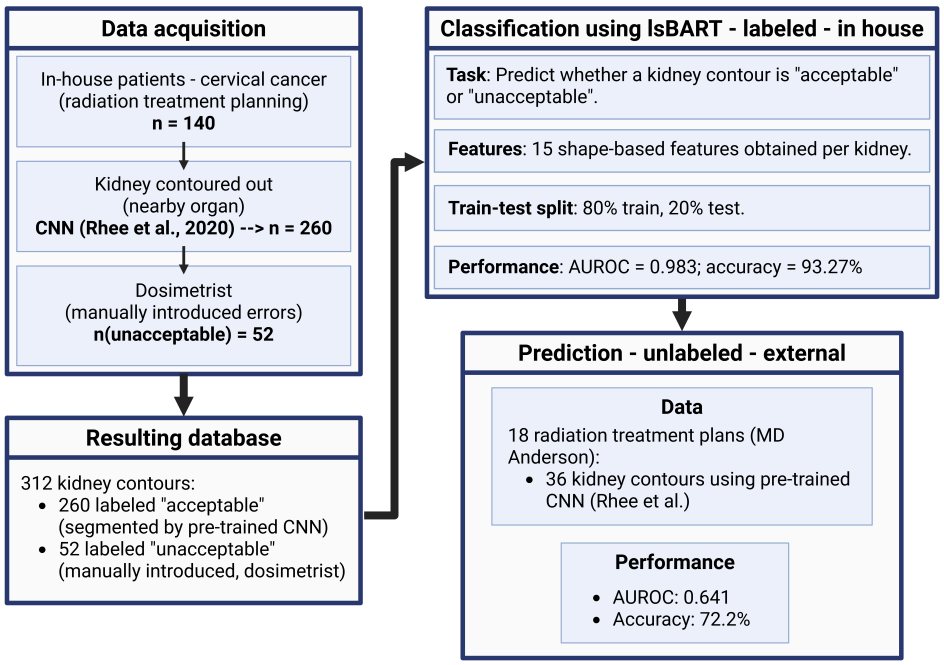


▼ **Step 5 — generating all functional predictors and corresponding noisy realizations:**

- All five functional predictors:
  $F(t) = \{f_1(t), f_2(t), f_3(t), f_4(t), f_5(t)\}$, where $f_1(t) = f(t)$ up until step 4 above;
- And, 10 associated noisy realizations pertaining to each
  $X = \{X_1(t), X_2(t), X_3(t), X_4(t), X_5(t)\}$, where $X_1(t) = X(t)$ up until step 4 $= f_1(t) + \varepsilon_{1,i}(t)\ \forall\ i \in \{1, 2, 3, \ldots, 10\}$.

5.1.2 Real-data case study — application to radiation treatment planning

▼ **5.1.2.1 Overall workflow**



5.2 Noise generation in the benchmarking paper

▼ Expand ***Section 5.2 The noise generation process***

5.2.1 Aim

To generate errors that resemble errors produced by AI-based segmentation models closely.

5.2.2 Methods

- **Setup**: Given an original organ segmentation mask, M, where: d is distance between the chosen point and the nearest boundary pixel on the original mask, r_{max} is the maximum radius of the original mask, and r is the distance between the new point and the centroid.

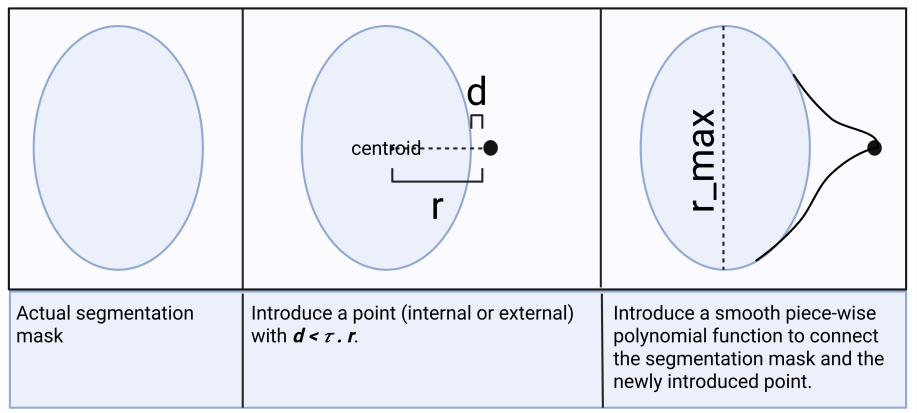
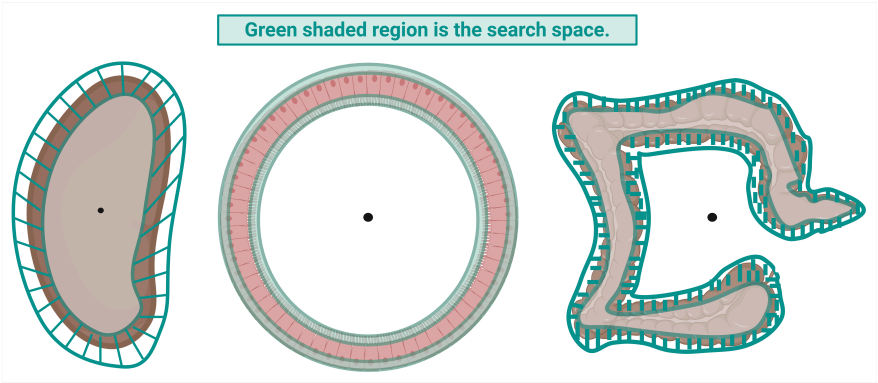


Fig. Selection of random point, that drives the direction of segmentation error.

- **Search space**: We try to either over-segment or under-segment the mask such that the random point that decides the direction of such over- or under-segmentation is bound by the search space, $\frac{d}{min(r,r_{max})} < \tau$.
  - This enables us to control the search space to a specific distance from the boundary of the actual mask – thereby incorporating both shape-preservation but also stochastic freedom within that search space.
  - As we increase τ, the search space increases – and it is more likely to find the random point farther away from the boundary.



  - **Case 1 — Under-segmentation**: In the case of under-segmentation (that is, point is inside of the mask region), the equation reduces to $\frac{d}{r} < \tau$.

- **Case 2 — Over-segmentation (that is, point is outside of the reference mask region):** In the case of over-segmentation (that is, point is outside of the mask region), the equation remains $\frac{d}{min(r,rmax)} < \tau$.

- **Hyperparameters**:
  1. Distance threshold, $\tau = \frac{d}{min(r,rmax)}$.
  2. The smoothing function-based hyperparameters, $\mathbb{S}$.

- **Why is restriction of search space important?**
  1. We want to preserve shape-based integrity of the segment → **Purely stochastic models with an unbound search space** are unable to do this.
  2. But, we also do not want to constantly induce **"more" errors in small organs & "less" errors in big organs** by fixing a **constant** search space. → **Shape-agnostic, fixed-bound models are unable to do this.**

- **Disadvantages of shape-agnostic models**
  1. Requires organ-specific processing.
  2. Constant need for quality analysis (QA), highly subjective, and very susceptible to these QA decisions.

     ⇒ Majority by area? What percentage of area? Information content? → We need to constantly make these decisions on an individual organ — if not sample — basis.

Code versioning and implementation details

▼ Expand ***Smoothing function version 12.11.2025***

12.11.2025, The corrected smoothing function —> `def directional_oversegmentation`

Step 1 [Same as before]:

The first step is for us to convert the `unint8` (or other) input slice to a `bool` binary mask.

```
mask = mask_slice.astype(bool)
if not np.any(mask):
    return mask_slice.astype(np.uint8)
```

⇒ if mask is empty (meaning, no organ present in that particular slice, then skip this slice and move onto next slice.

Step 2 [Same as before]:

Organ boundary.

```
boundary = binary_dilation(mask) & (~mask)
```

Step 3 [Same as before]:

Randomly select a pixel, from "limited" search space.

```
coords = np.column_stack(np.where(boundary))
idx = np.random.randint(len(coords))
y0, x0 = coords[idx]
```

Step 4 [New — updated] I produce a "small" bump, ***bump_field***, like so:

`bump_field = np.zeros_like(mask, dtype=float)` → this is the empty bump field, before bump induction.

After bump induction:

`bump_field[y0, x0] = max_offset_px` → max_offset_px is a hyperparameter that controls the size of the bump.

Step 5 [New — updated] Gaussian smoothing of the bump:

`bump_field = gaussian_filter(bump_field, sigma=smooth_sigma_px)`

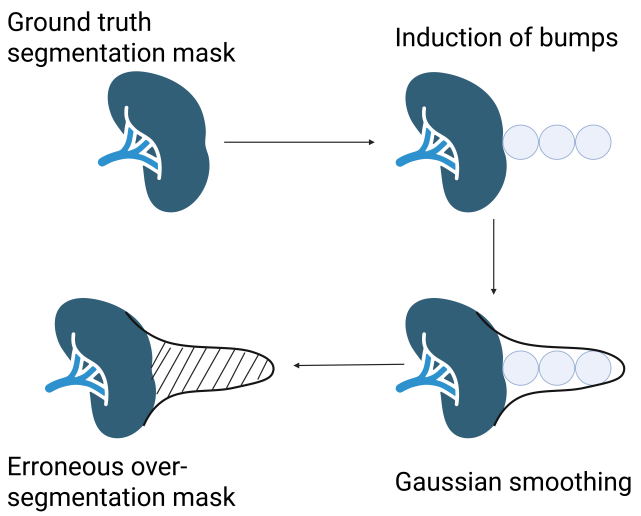
Why do we do this:

The reason that the bump was "spliny" in the last version was because we were actually not doing gaussian smoothing at all. We were just using a piece-wise polynomial function to smoothen the boundary of the many generated noisy circles so that the over-segmentation does not look jagged.

→ But, smoothness of over-segmentation noise does not ensure smoothness of the ground truth organ segmentation.

Piece-wise smoothing function

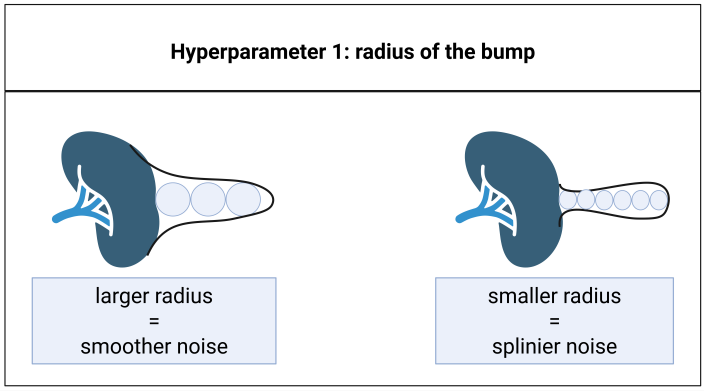
Although the new, Gaussian-smoothed version requires no piece-wise smoothing (because it is already a smooth function — it has been Gaussian smoothed), we are still retaining the piece-wise smoothing function. **[It is redundant, but we retain it for good measure.]**



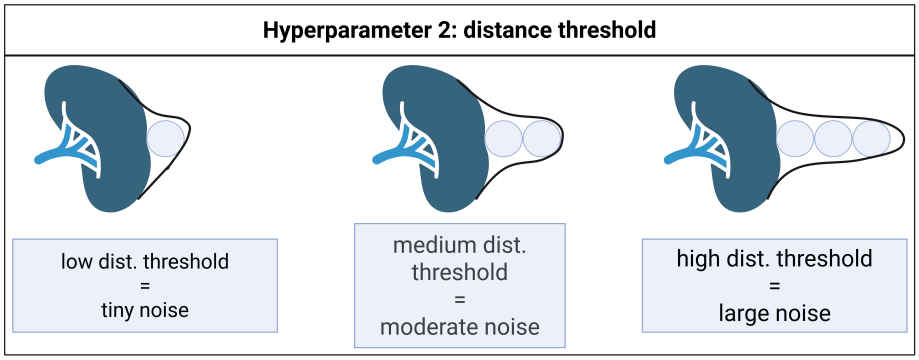
Therefore, the smoothing function contains two hyperparameters:

1. Radius of bump (determines smoothness/ spliny-ness of the over-segmentation), $\$$
2. Distance threshold (determines size of oversegmentation), τ

1. Radius of bump, $\$$



2. Distance threshold, τ



▼ Expand ***Smoothing function version 11.26.2025***

11.26.2025, The original smoothing function —> `def directional_oversegmentation`

Creates over-segmentation by joining a tubular structure (which is a combination of noise-induced circles close together) between the boundary of the organ and a point:

1. Internal point — undersegmentation
2. External point — oversegmentation
3. Mixed under- + over-segmentation

→ But first, we need to constrain the search space that we want to search the target point in. That is done by the `def select_target_point_with_constrained_search_space` function. This function has been described later on.

- Once we select a random point as so:

```
target_point = select_target_point_dr_constrained(
slice_mask,
tau=0.3   # dist. threshold
)
```

- Once we have a selected point, we create the segmentation error, e.g. oversegmentation using `def directional_oversegmentation` :

```
noisy_mask = directional_oversegmentation(
slice_mask,
mode="tube",
target_point=target_point
)
```

- Steps involved in directional over-segmentation:
  - Copy the mask:

    `mask = mask.copy()`
  - Extract coordinates of all pixels contained within the boundary of the mask:

    `mask_coords = np.argwhere(mask)` ⇒ we will call them “organ pixels” for the remainder of this document.

    ```
    if not np.any(mask):
      return mask_slice.astype(np.uint8)
    ```
    ⇒ if mask is empty (meaning, no organ present in that particular slice, then skip this slice and move onto next slice.
  - Find mask pixel closest to target point — we can visualize this as the point on the organ that is facing towards the point:

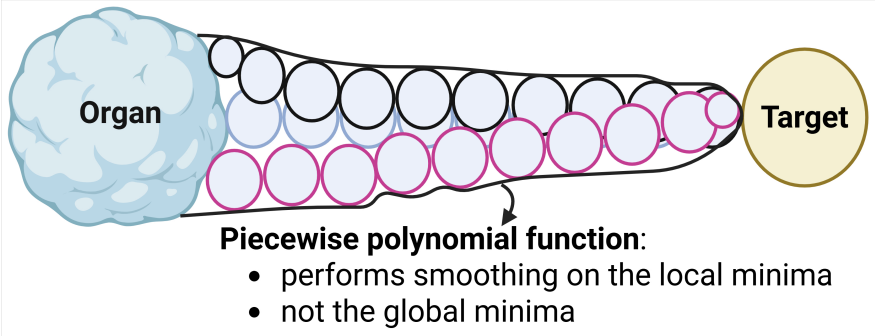
    ```
    distances = np.linalg.norm(mask_coords - target_point, axis=1)
    start_pixel = mask_coords[np.argmin(distances)]
    ```
  - Calculate the actual vector depicting direction from the point on the organ to the point chosen randomly:

    ```
    vec_y = y1 - y0
    vec_x = x1 - x0
    norm = np.linalg.norm([vec_y, vec_x])
    ```
  - Normalize by distance:

    ```
    vec_y /= norm
    vec_x /= norm
    ```
  - Move from source to target:

    ```
    for r in range(int(norm)):
    cy = int(round(y0 + vec_y * r))
    cx = int(round(x0 + vec_x * r))
    ```
  - At each step, a circle (with a certain degree of random noise — hyperparameter that can be set by the user) is added (radius of disk is also a hyperparameter — `max_radius`):

    ```
    rr, cc = np.ogrid[-max_radius:max_radius+1, -max_radius:max_radius+1]
    disk_mask = rr **2 + cc** 2 <= max_radius**2
    mask[y_min:y_max, x_min:x_max] |= disk_mask[...]
    ```



- Detailed noisy masks on hundreds of samples across five organs:
  - Two easy-to-segment organs — liver, kidney.
  - Three difficult-to-segment organs — prostate, esophagus, inferior vena cava.

⇒ Results can be found here:

> Organ-wise-over-and-under-segmentation-results.pptx

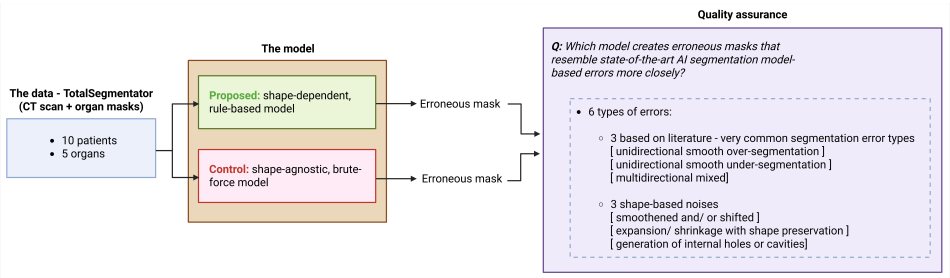
- Feedback:
  - “Under-segmentations looks perfect!”
  - Mixed over- and under-segmentation is unrealistic when:
    - We have more than two splines.
    - We have holes within the organ.
      - Most state-of-the-art organ segmentation tools have rule-based algorithms that fill in holes within organ segmentations.

        ⇒ Get rid of the Swiss cheese model basically — that is unrealistic of AI model-based errors.
  - Over-segmentation is sometimes “too spliny.” Specifically, the protrusion must be less pronounced.

▼ Expand ***Noise generation version 1***

In the first version of the noise generation function, I identified 6 types of errors based on literature.

Method



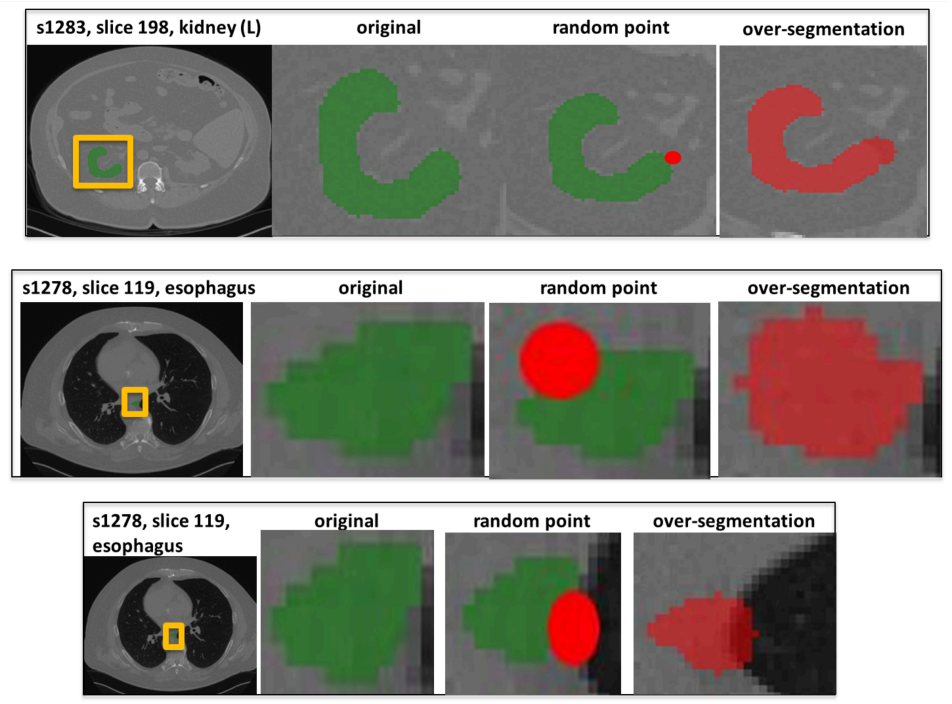
Disadvantages

- Error types are very “intuitive,” not really grounded in “good” literature.
  - There is not enough literature out there that systenatically benchmarks 3D errors made by AI-based segmentation models.
  - There is some, albeit very limited, state-of-the-art benchmarking studies on 2D errors (Saporta et al., 2022 [12]) — but even fewer on 3D errors, and errors made by AI-based segmentation models that perform organ segmentation in 3D.

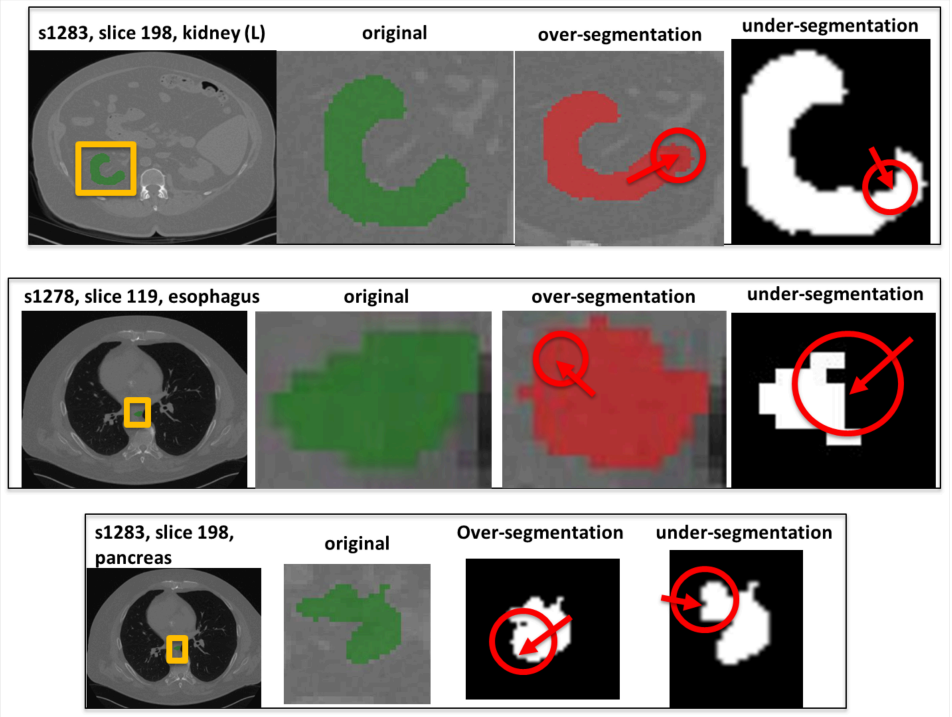
- Selection criteria
  - **10 patients**: 10 patients chosen randomly from $\{\text{chest}, \text{thoracic}\}$ CT scans.
  - **5 organs**: 3 difficult to segment, 2 easy to segment — based on literature. ***(For more details, see Section 3.2 “IsBART was only trained on one organ — that too, an easy organ to segment by AI-based models, the kidneys”.)***
  - **6 types of errors**: 3 types of noises based on literature; and 3 types of noises based purely on shape. ***(For more details, see Section 3.1 “IsBART was not trained to be able to classify true organ segmentation versus AI model-based errors”.)***

5.2.3 Types of noise generated

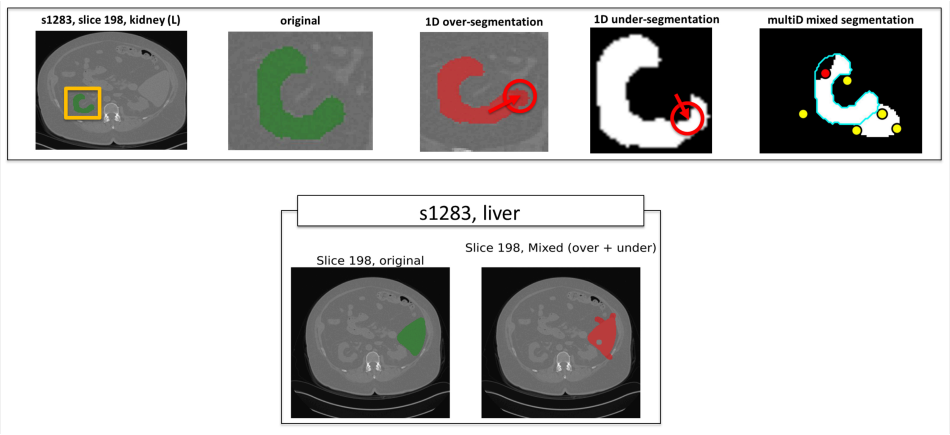
5.2.3.1 Unidirectional, smooth over-segmentation



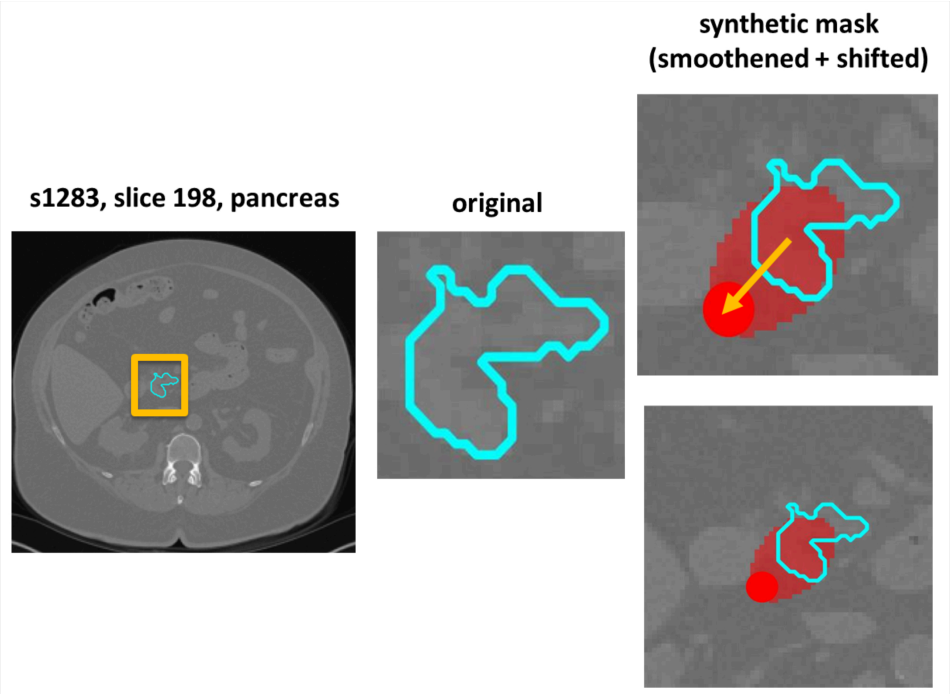
5.2.3.2 Unidirectional, smooth under-segmentation



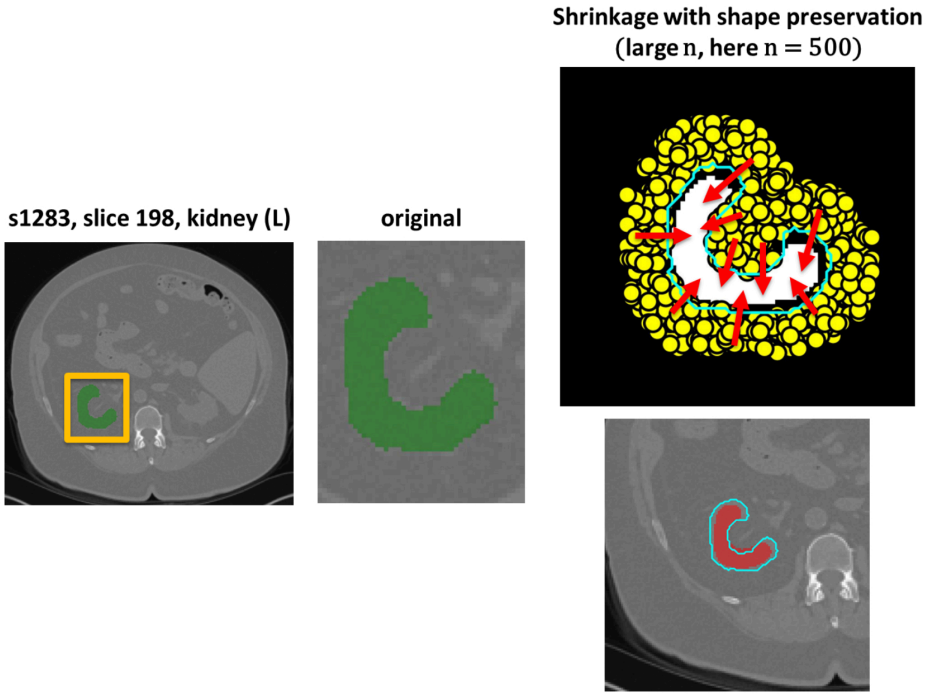
5.2.3.3 Multidirectional, mixed segmentation (over + under)



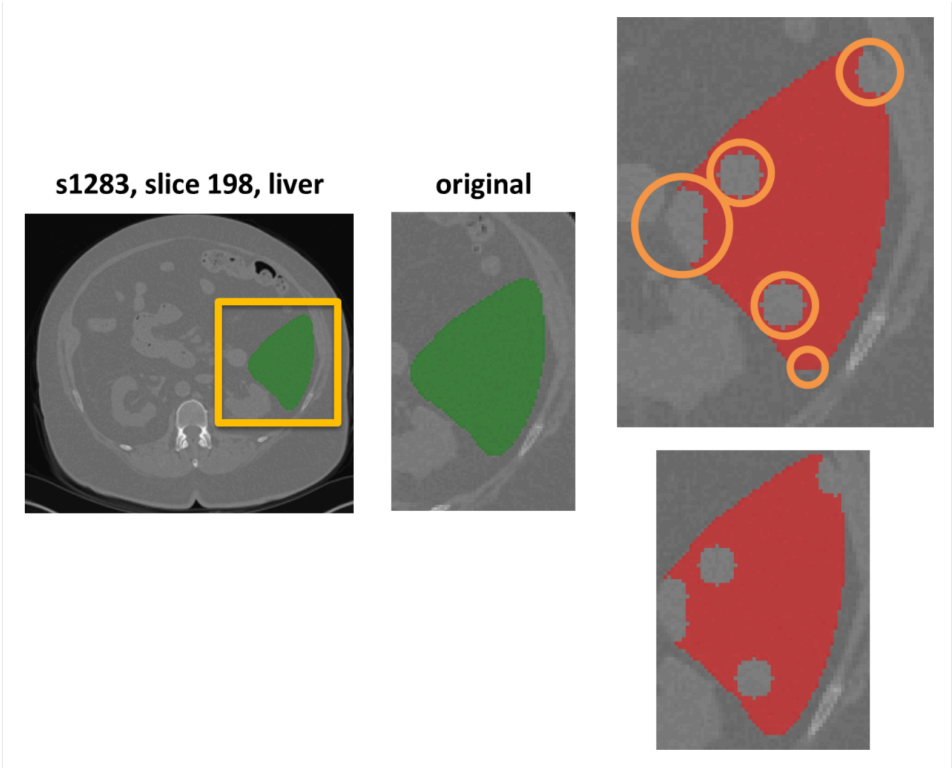
5.2.3.4 Shape smoothening and/ or shifting



5.2.3.5 Expanded/ shrunk with shape preservation



5.2.3.5 Generation of internal holes or cavities



Feedback

- The literature is not robust — just stick with over- and under-segmentation.

Weekly presentations

▼ **12.11.2025**

Presentation slides

Organ-wise-over-and-under-segmentation-results.pptx

Feedback

1. While looking at the 3D renderings of the noises generated in 2D, they do not realistically mimic AI model-based segmentation errors. Actually run trained TotalSegmentator model on AMOS, BTCV and FLARE.
2. Generate protocol for noise characterization - short (less than 3 pages)
3. [Later] No SIIM abstract because we do not have results — but I decided I would create one anyway, and run it by Paul — as a practice run for the text only.

▼ **11.26. 2025**

Presentation slides

Organ-wise-over-and-under-segmentation-results.pptx

error-characterization-protocol.pptx

Feedback

- Over-segmentation errors look too spliny. Make them:
  - Smoother
  - Shorter as compared to the breadth of the actual organ in 2D
- Under-segmentation is fine.
- Mixed (over- + under-) segmentation and the previously named “Swiss cheese” error type are not very representative of deep learning-based segmentation errors. Should just not make those errors any longer.

▼ **The meeting before that**

Presentation slides

manual1_IsBART.pdf

▼ **The meeting before that**

Presentation slides

PROTOCOL_IsBART_induced_noise.pdf

References

▼ Expand ***References***

1. https://bmcmedimaging.biomedcentral.com/articles/10.1186/s12880-022-00825-2
2. https://link.springer.com/article/10.1007/s40747-024-01751-2
3. https://ieeexplore.ieee.org/abstract/document/10609512?casa_token=22s182eLWLYAAAAA:WkUroKa51fcSfRialmsBqI4UxXhxDekhymygzkSgU72b8qY2hnZLcRUxzJnjeL-ypDs8Qw2fyukB9zI
4. https://insightsimaging.springeropen.com/articles/10.1186/s13244-025-02098-z
5. https://arxiv.org/abs/2309.13675
6. https://link.springer.com/article/10.1007/s00234-024-03429-5
7. https://www.frontiersin.org/journals/oncology/articles/10.3389/fonc.2024.1328146/full
8. https://www.mdpi.com/2076-3417/13/1/329
9. https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-022-06096-y
10. https://link.springer.com/article/10.1007/s13755-023-00220-3
11. https://www.mdpi.com/2306-5354/11/5/427
12. https://www.nature.com/articles/s42256-022-00536-x

▼ **Ideas and/ or work in progress**

MODEL AND DATASETS- “Case Study”:

1. Describe the operations for quantifying the metrics defined in the Framework, and how you plan to analyze and summarize them. This includes descriptive statistics plan and comparative statistical plans. Consider making mock tables or graphs to illustrate what these would look like. **The goal is for the PIs to understand what you have planned and to identify if there are fundamental holes or gaps that we need to address prior to doing the study.**

Next, we formulate a deterministic and completely quantitative metric for measuring the extent of over- and under-segmentation. We call this metric error score, β, which is defined as the difference between the area of the predicted segmentation and the ground truth segmentation, normalized by the area of the ground truth segmentation: $\beta = \frac{Area(predicted) - Area(actual)}{Area(actual)}$. A positive value of β is indicative of over-segmentation, whereas a negative value of β is indicative of under-segmentation.

Next, we extract shape-based features from the ground truth segmentation. Subsequently, we perform correlation between the shape-based features and the segmentation score, for each of the five organs separately — in order to

elucidate specific shape-based characteristics that make organs susceptible to certain kinds of errors. Specifically, this is done by fitting a simple linear regression model, with each of the features as separate independent variables, and the error score β as the dependent variable. The regression coefficient of the fitted simple linear regression would tell us how each changing feature affects the extent of over- or under-segmentation. Figure 1 provides the reader with a pictorial overview of the workflow.

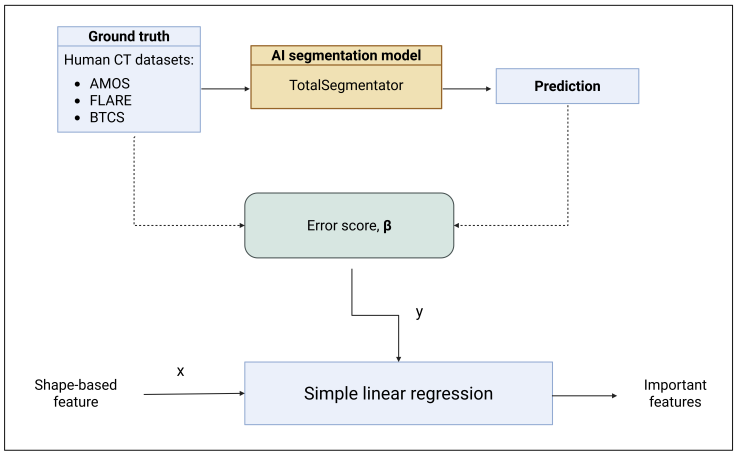


Fig 1. A pictorial overview of the error characterization workflow.

3. Methods

3.1 Run TotalSegmentator on AMOS, BTCV and Flare — for liver, kidneys, pancreas, esophagus, inferior vena cava.

▼ Expand ***Section 3.1***

3.2 Formulation of a deterministic and completely quantitative metric for measuring the extent of over- and under-segmentation.

▼ Expand ***Section 3.2***

3.3 Fitting a simple linear regression model, with each of the features as separate independent variables, and the error score β as the dependent variable.

▼ Expand ***Section 3.3***

The regression coefficient of the fitted simple linear regression would tell us how each changing feature affects the extent of over- or under-segmentation.

▼ **Weekly presentations — 2026**

▼ Jan 08, 2026

error-characterization-protocol-jan-08-2026.pptx