CT Organ Localization with Reinforcement Learning

Group Member: Kexin Yang, Hainan Xiong, Haoming Chen

Mentor: Alex Chowdhury

Background and Motivation

Identifying organ location in CT scans is an important but time-consuming task for many medical tasks such as disease diagnoses. So Deep Learning algorithm may be extremely helpful to improve radiologists' working efficiency if it can accurately localize the organ of interests as a pre-processing step.

Problem:

- Most of current SOTA model require large labelled datasets
- Require the model to have relatively high accuracy before the model can be deployed in real-world situations

Project Goal

We want to design a reinforcement learning algorithm that can automatically localize specific organ(s) in CT images to improve the working efficiency of the disease diagnosis progress.

- Baseline: a reinforcement learning-based organ localization model proposed by Navarro et al. (2020)
- Goal: train the model with relatively **small number of labeled data** using RL-based model and try to **improve the model performance**
- Evaluation Metrics: 3D IoU score (intersection over Union)

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ Union}$$

Dataset

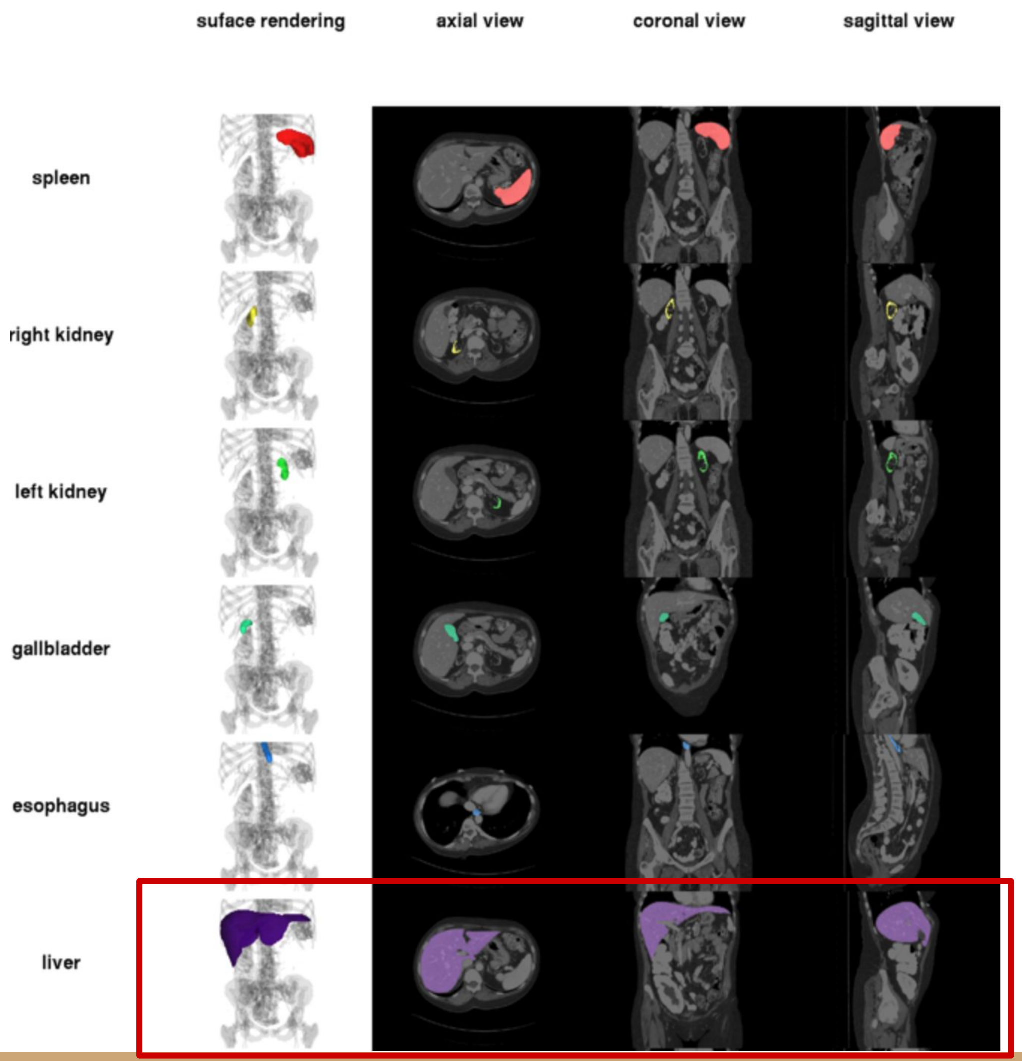
We decided to use a publicly available Multi-Atlas Labeling CT Dataset. It contains **50 ab domen CT scans** randomly selected from a combination of ongoing colorectal cancer chemotherapy trials and a retrospective ventral hernia study. The 50 scans were captured during portal venous contrast phase with variable sizes:

- Volume sizes: 512 x 512 x 85 - 512 x 512 x 198
- Field of views: 280 x 280 x 280 mm3 - 500 x 500 x 650 mm3
- In-plane resolution: 0.54 x 0.54 mm2 - 0.98 x 0.98 mm2
- Slice thickness: 2.5 mm to 5.0 mm

Dataset

We decided to use a publicly available Multi-Atlas Labeling CT Dataset, which contains **30 abdomen CT scans** with labels of 13 different organs.

- We splitted the dataset into **24 training samples + 6 testing samples**
- For this project, we will start with **liver localization**



Organ Labels

The label for the Dataset contains **13 abdominal organs** that were manually labeled by two experienced undergraduate students and verified by a radiologist on a volumetric basis using the MIPAV software, including:

(1) spleen

(2) right kidney

(3) left kidney

(4) gallbladder

(5) esophagus

(6) liver

(7) stomach

(8) aorta

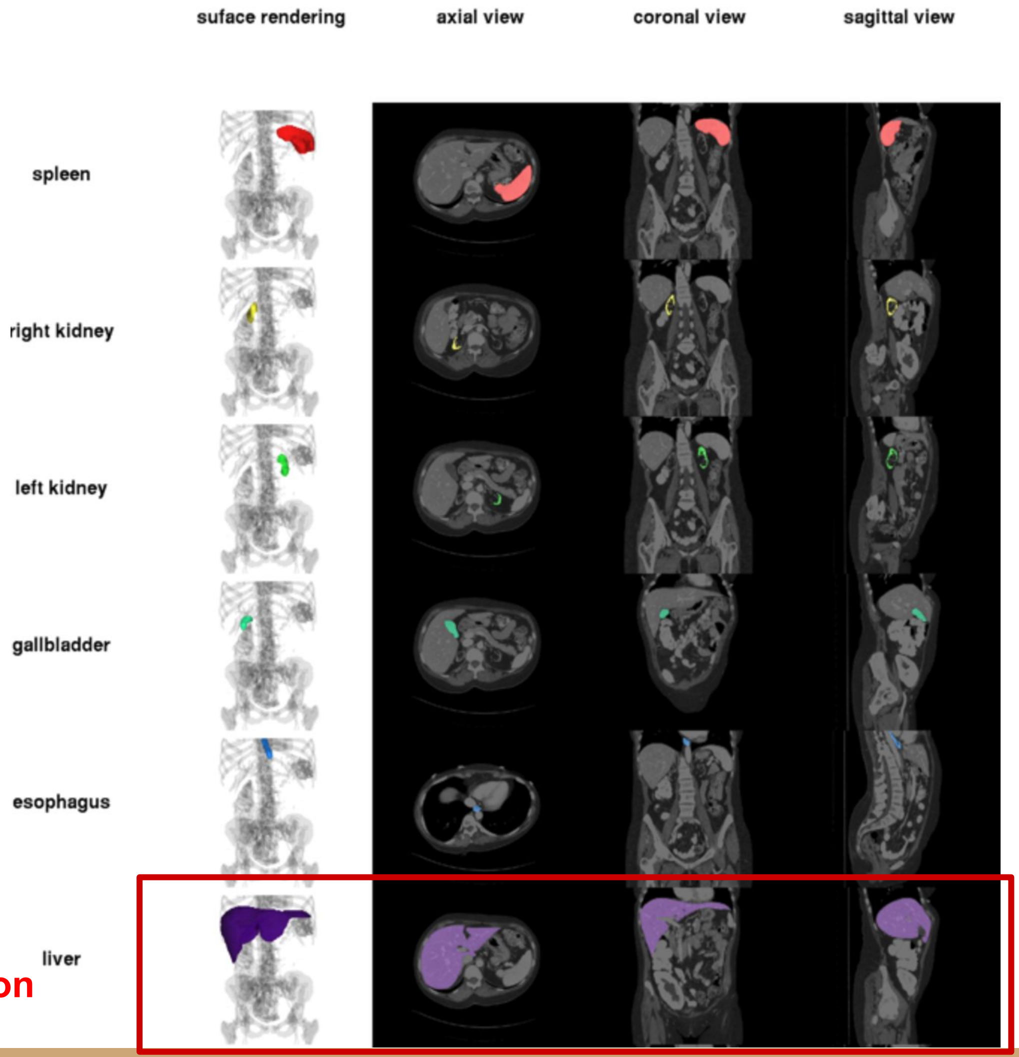
(9) inferior vena cava

(10) portal vein and splenic vein

(11) pancreas

(12) right adrenal gland

(13) left adrenal gland



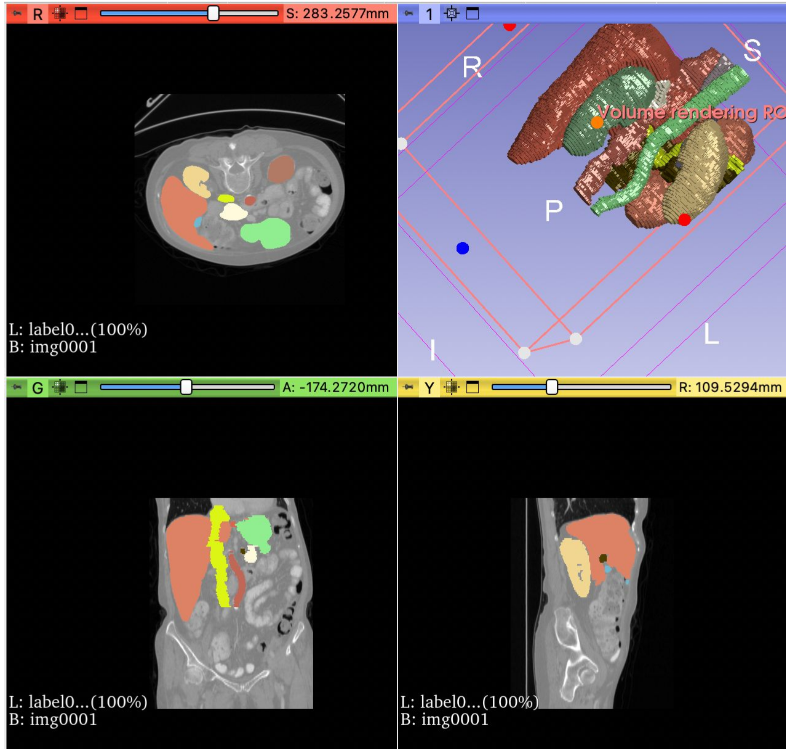
For this project, we will start with liver localization

Dataset

Training Dataset: There are **30 total training data** that each contains a 3-dimensional CT scan of the abdomen. A corresponding label is provided, which contains a 3-dimensional segmentation for multiple organs of interest.

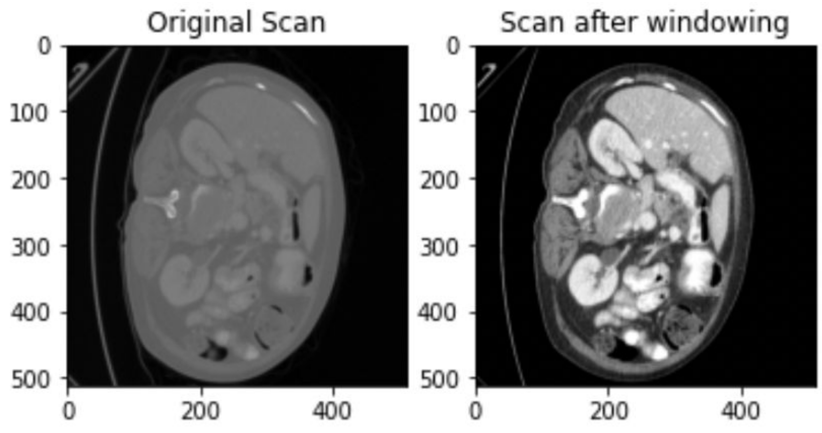
Testing Dataset: There are **20 testing data**, which also contains a 3-dimensional CT scan of the abdomen. No label is provided.

Limitation: One limitation would be **the generalizability of our model performance in other datasets with varying CT scan volume and resolution.** Even though our Dataset does contain some variability in terms of volume and resolution, it clearly doesn't cover all possible ranges and therefore may not generalize to other Datasets.



Data Pre-Processing

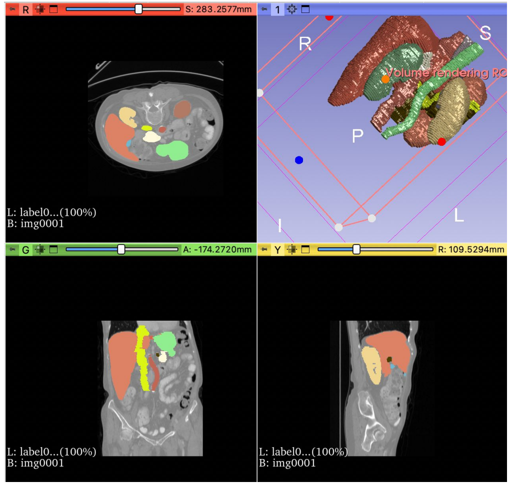
Step 1: CT Scan Windowing

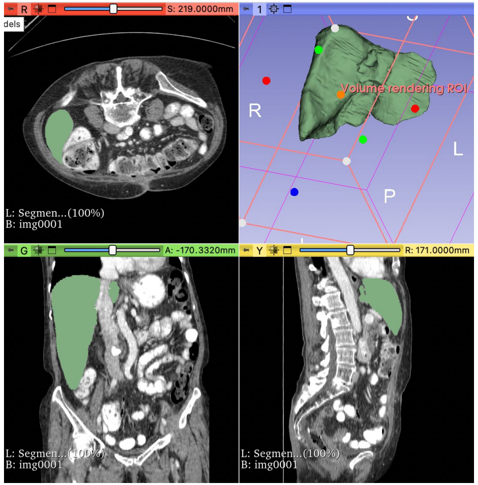


Help highlight key anatomy so that the images can be analyzed easily.For our project, the ideal pixel range for abdomen region is <u>-160 to 240</u>.

Step 2: Select Target Label Class

Start with liver localization





Step 3: Bounding Box Label Generation

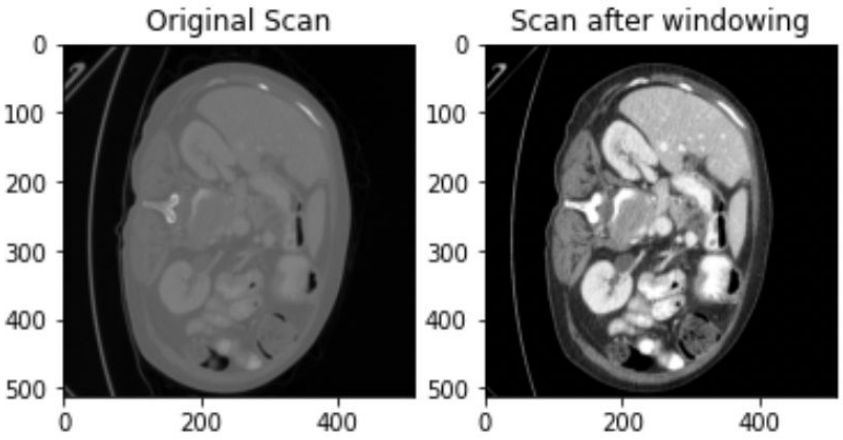
Data Pre-Processing

Step 1: CT Scan Windowing

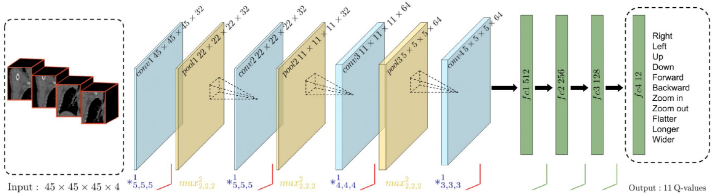
The primary type of preprocessing used for CT scans is windowing, which helps highlight key anatomy so that the images can be analyzed easily. This is because certain organs will be most clear at certain pixel intensities due to certain characteristics such as their density. Therefore, we can choose an ideal pixel value range to highlight the organ of interest.

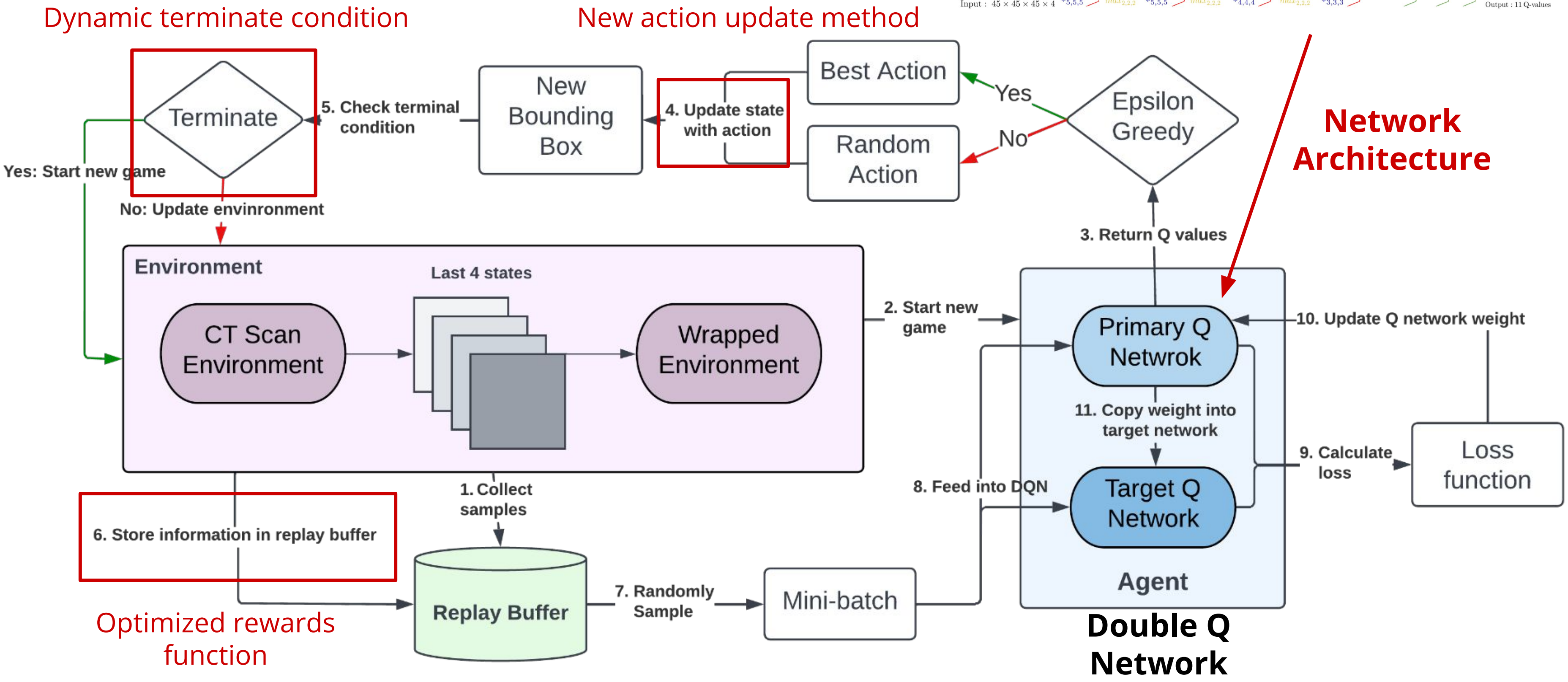
Ideal pixel range for abdomen region: -160 to 240

- Window Center: 40
- Window Width: 400



Model Framework





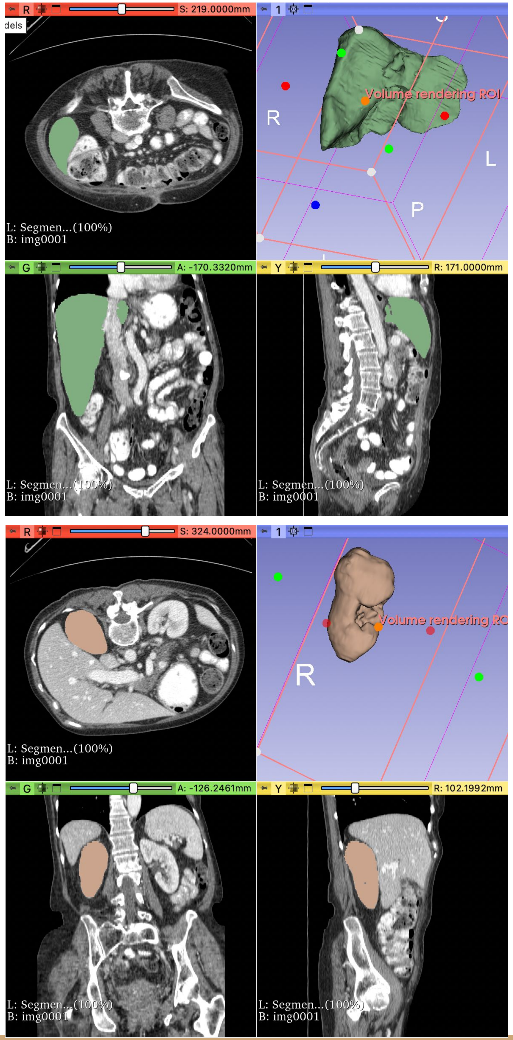
Data Pre-Processing

Step 2: Select Target Label Class

We will train each model to localize one single organ, so in the data pre-processing step, we will only extract label mask associated with the one organ. For our project, **we will start with liver (label = 6)**, because the liver is typically the easiest to segment due to the large volume and usually a homogenous, ellipsoidal shape.

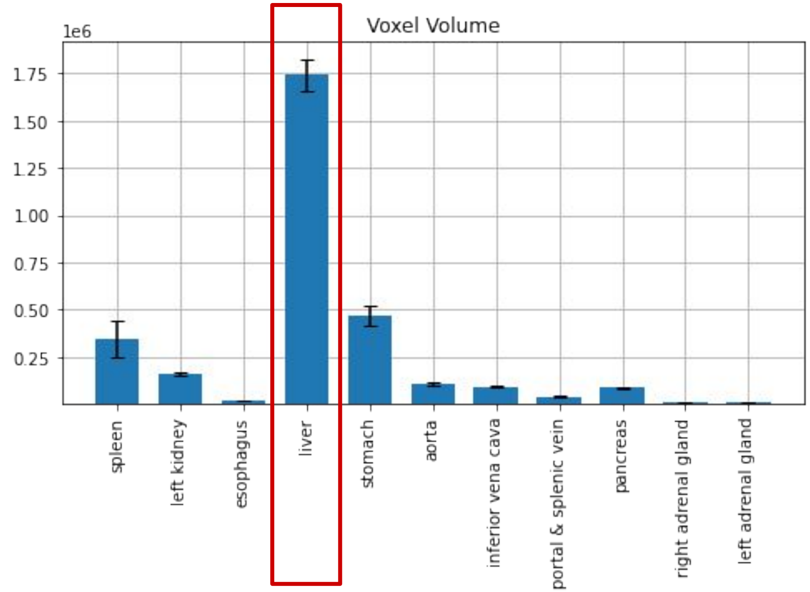
Step 3: Bounding Box Label Generation

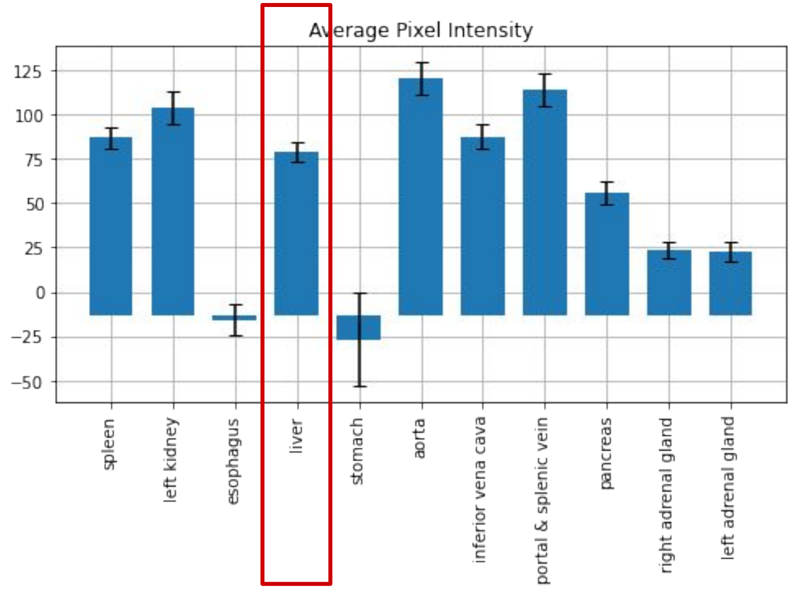
The original organ label is provided as a segmentation mask, where the value indicates the type of organ (1 - 13). For our model, we will use the *generate_spatial_bounding_box()* function from *monai.transforms.utils* module to transform the segmentation mask into a bounding box that is represented by its upper-left-front coordinate (x, y, z), and lower-right-back coordinate (h, w, d).



Exploratory Data Analysis

Since we do not need to generate any feature for our model, we simply plotted the **voxel volume** and **average pixel density** for different types of organ. **The purpose was to choose which organ to start with**, since we want to focus on one specific organ that is relatively easy to localize in order to quickly debug and tune our model.





Innovations

	Baseline Model	Our Model
Rewards Function	● Binary Rewards: -1, +1	● Proportional to change in IoU score ● Additional terminal rewards ● Penalty for out-of-boundary action
Terminal Condition	● Oscillation occurrence ● Or reaches terminate IoU: fixed to be 0.85	● Max step size, increased with more training epochs: 15 to 20 ● Or reaches terminate IoU, increased with more training epochs: 0.60 to 0.85
Action Update Method	● Proportional to bounding box length	● Proportional to bounding box length ● Set minimum and maximum threshold ● Handle out-of-boundary cases

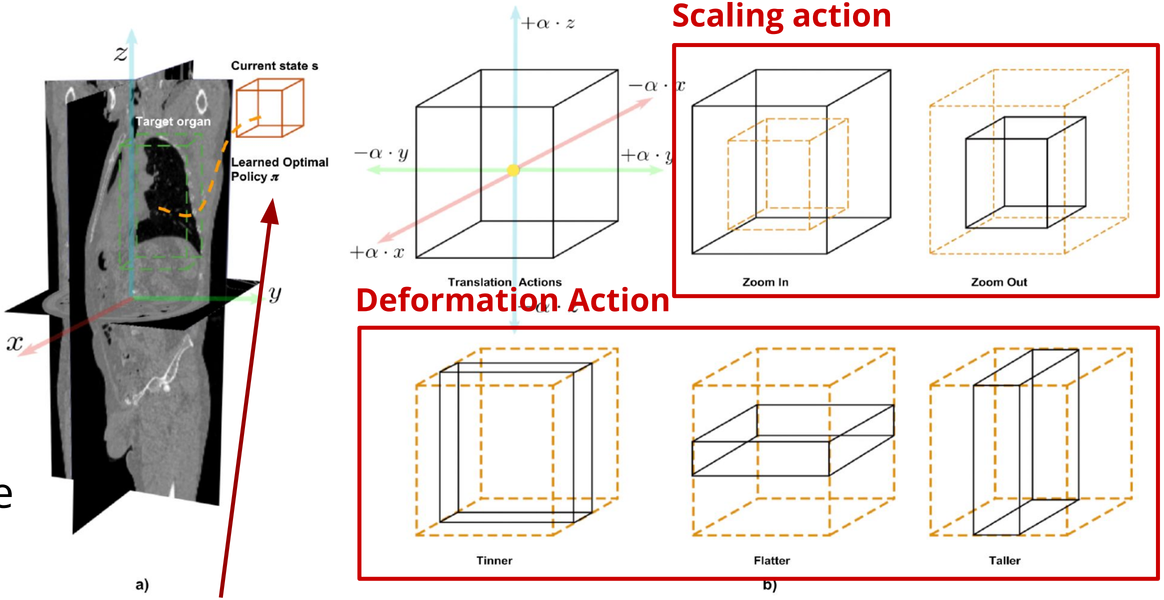
Model Framework - Environment & Agent

Environment: 3D CT scan
State: voxel values in the current bounding box
Action: 11 possible actions

- 6 Translation action
- 2 Scaling action
- 3 Deformation action

Update Method:

- Step size: proportional to the size of the bounding box
- Minimum step size: 1
- Maximum step size: 10
- Handle out of boundary case



1. Start with some bounding box inside the CT scan

2. At each step, choose an action to move the bounding box, which ideally makes it closer to the true organ location

Model Framework - Rewards & Loss

Optimized Rewards Function

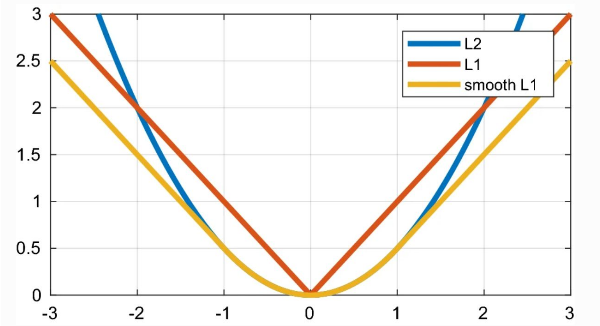
The baseline model uses a simple binary rewards to indicate increased vs. decreased IoU score.

- Reward is calculated as the difference between current and previous IoU times a scaling factor to reflect the degree of improvement
- Give additional rewards if terminal condition is reached within max step size
- Give penalty if the new bounding box exceeds the boundary

Loss

Smooth L1 Loss: A combination of L1 and L2 loss

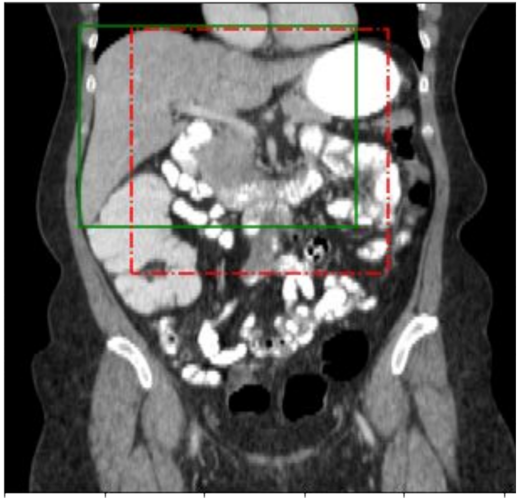
$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$



Example - Rewards Calculation

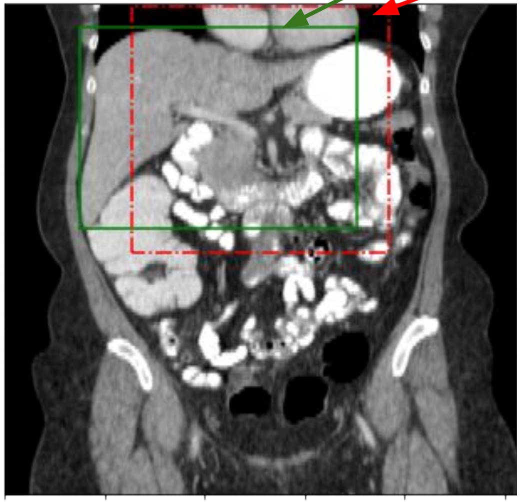
True Bounding Box Label

Model Predicted Bounding Box

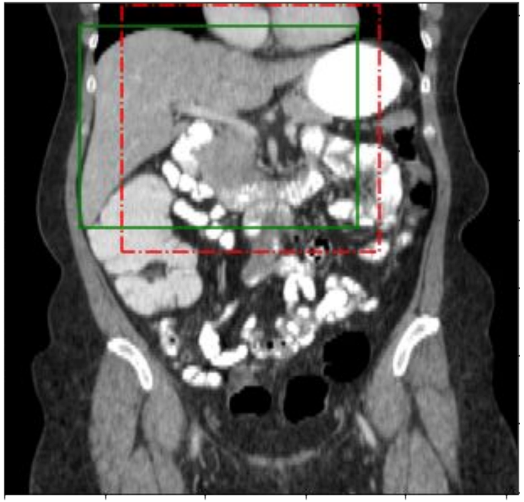


State 0
Initial IoU: 0.554
Terminal IoU: 0.80

Step size: 10% x box length

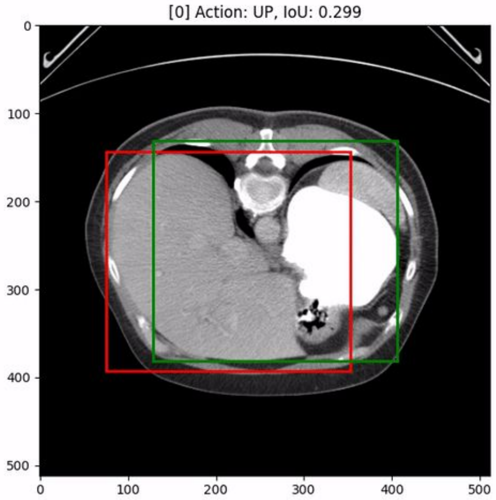


State 1
Action: move up
Terminal: False
Out of boundary: True
New IoU: **0.569**
Rewards: **(0.569 - 0.554) x 100 - 10 = 5**

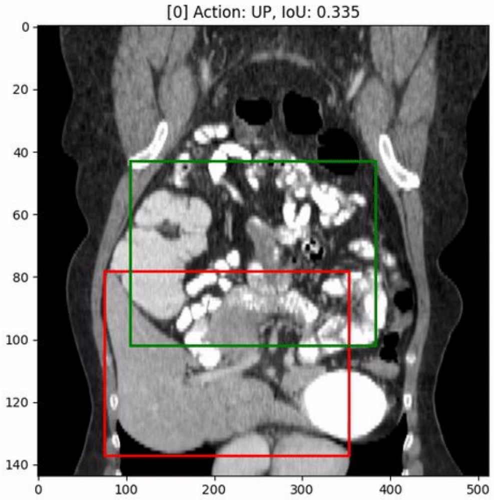


State 2
Action: move left
Terminal: False
Out of boundary: False
New IoU: **0.610**
Rewards: (**0.610 - 0.569) x 100 = 41**

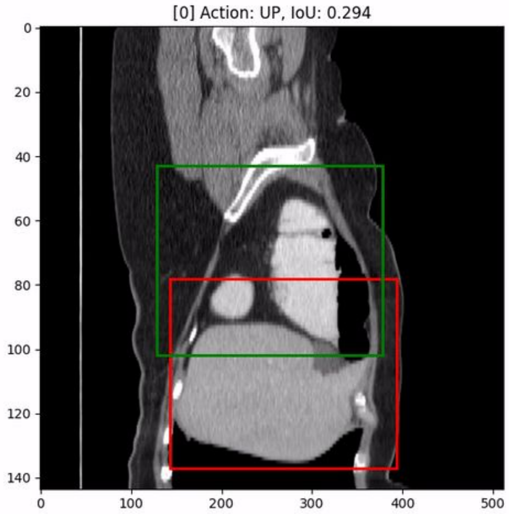
Example - Game Trajectory



Axial View



Coronal View



Sagittal View

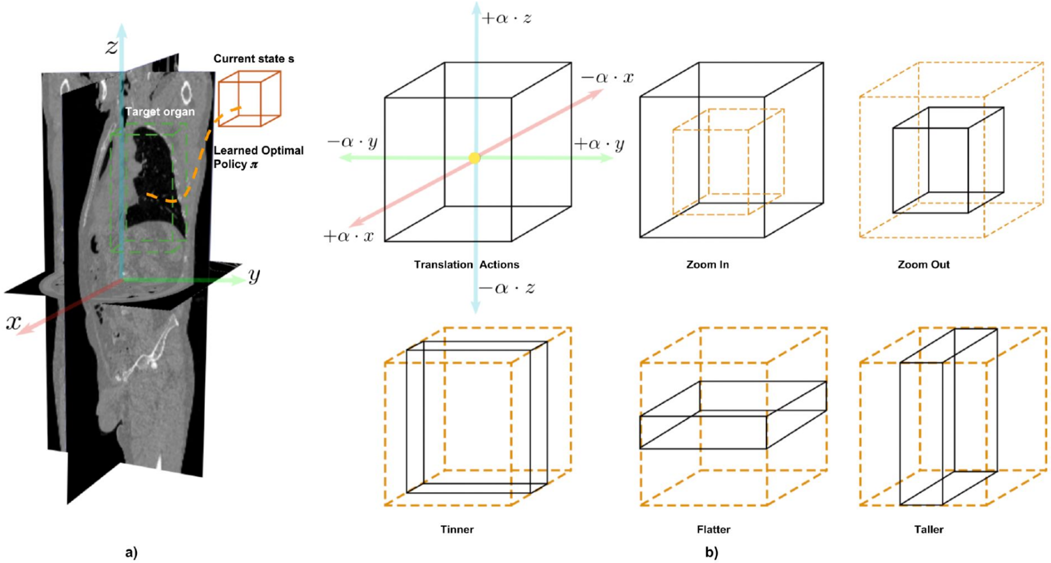
Baseline Model - Environment & Agent

Environment: the 3D CT scan

State: voxel values in the current bounding box

Action: 11 possible actions

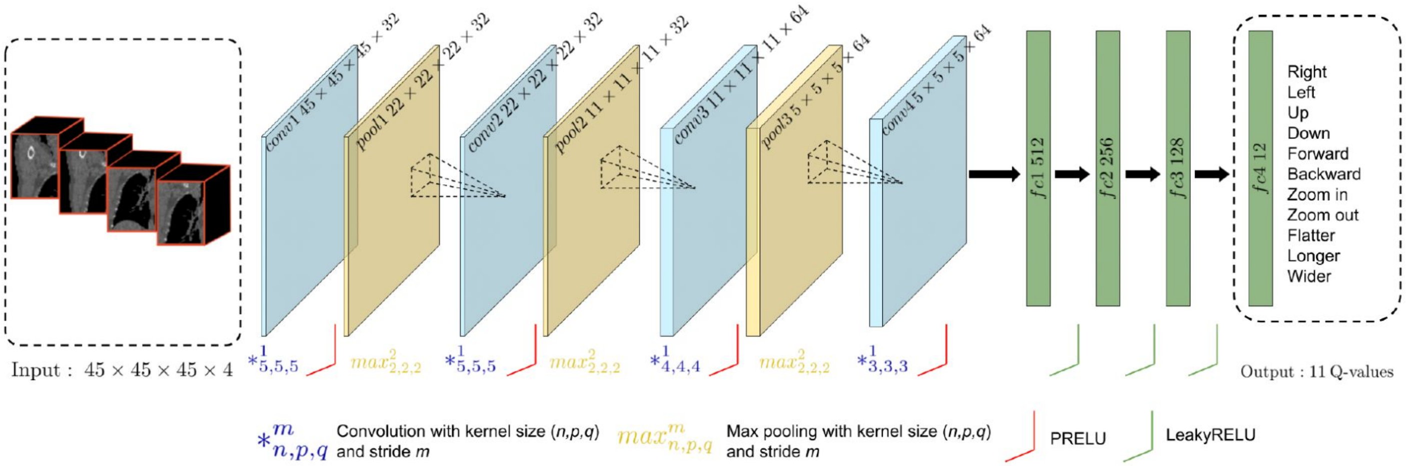
- Translation: x+, x-, y+, y-, z+, z-
- Scale: s+, s-
- Deformation: dx (thinner), dy (flatter), dz (taller)



Reward:

$$R_a(s, s') = sign(IoU(b', g) - IoU(b, g))$$

Model Framework - Deep-Q Learning



$$Q_{i+1}(s,a) = \mathbb{E}[r + \gamma \max_{a'} Q_i(s',a') \mid s,a]$$

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s',a')\sim U(D)}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2\right]$$

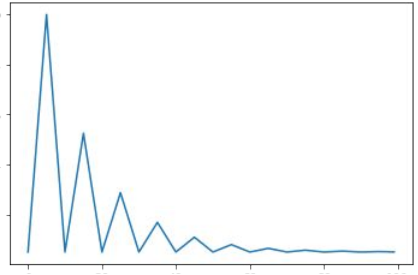
Baseline Model - Replay Buffer

One of the **main advantage of using RL-based model** is its ability to achieve good performance with limited labeled Dataset. This is largely due to the use of replay buffer.

- Stores trajectories of experiences that interact with the environment when executing a policy
- Sample these trajectory during training (possible in batch) to learn and optimize the agent’s policy
- Agents can interact with the same environment (labeled sample) many times and generate different trajectories form which it can learn

Training Config

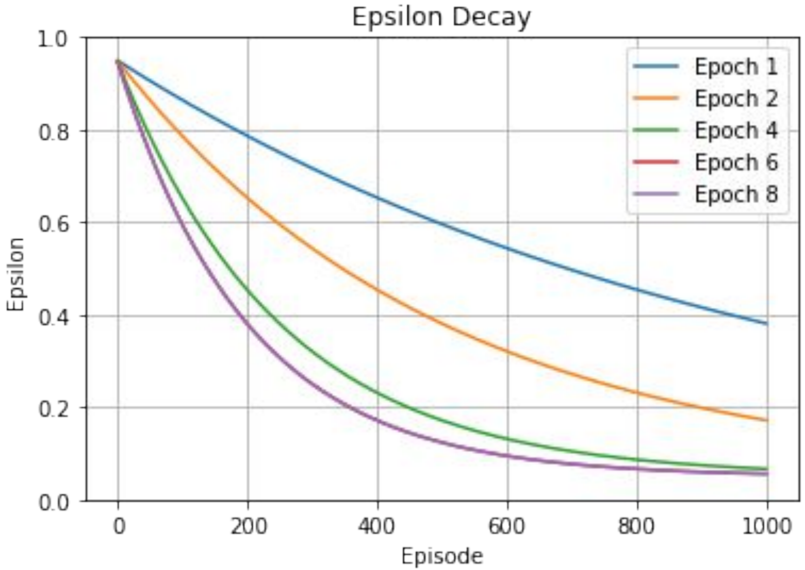
Optimizer:
AdamW
Learning Rate Scheduler:
Cyclic Annealing



	Baseline	Our Model
Training Data Size	70	24
Training Epochs	30	8
Batch Size	48	8
Replay Buffer Size	14,000	1,000

Epsilon: controls the probability of random action vs. best action

- Exponential decay to transition from exploration to exploitation
- Accelerate the decay to allow for more exploitation as the model gets more training

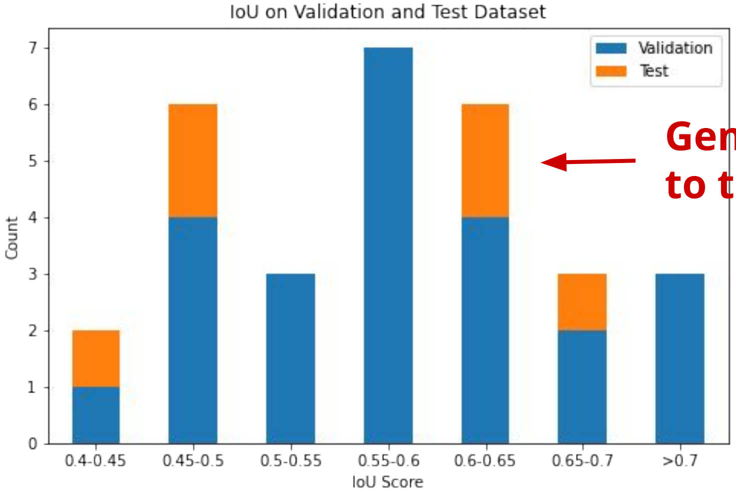


Model Performance

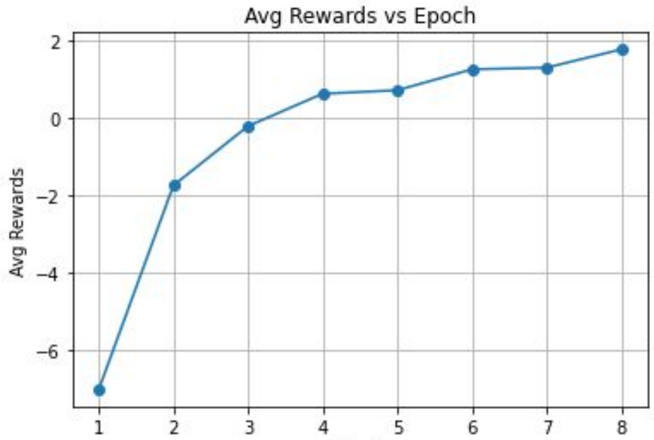
Baseline Test IoU (30 epochs): 0.80

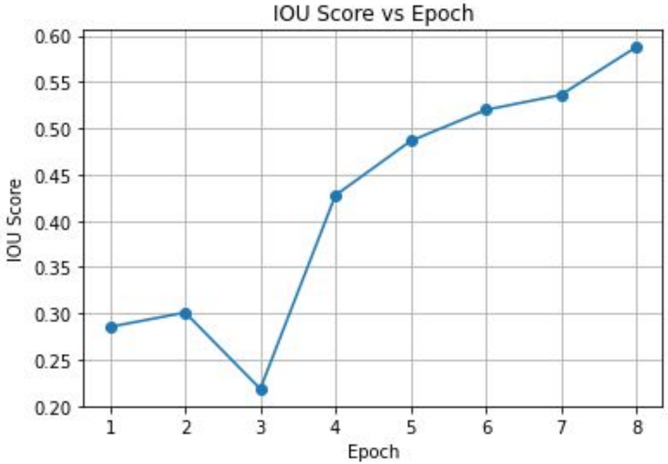
Our Model Validation IoU (8 epochs): 0.59

Our Model Test IoU (8 epochs): 0.56



Generalize well to test dataset





We expect the model performance to continue increasing with more training since the IoU score and rewards is clearly not converging after 8 epochs of training.

Example Bounding Box Prediction

True Bounding Box Label

Model Predicted Bounding Box

Best Example

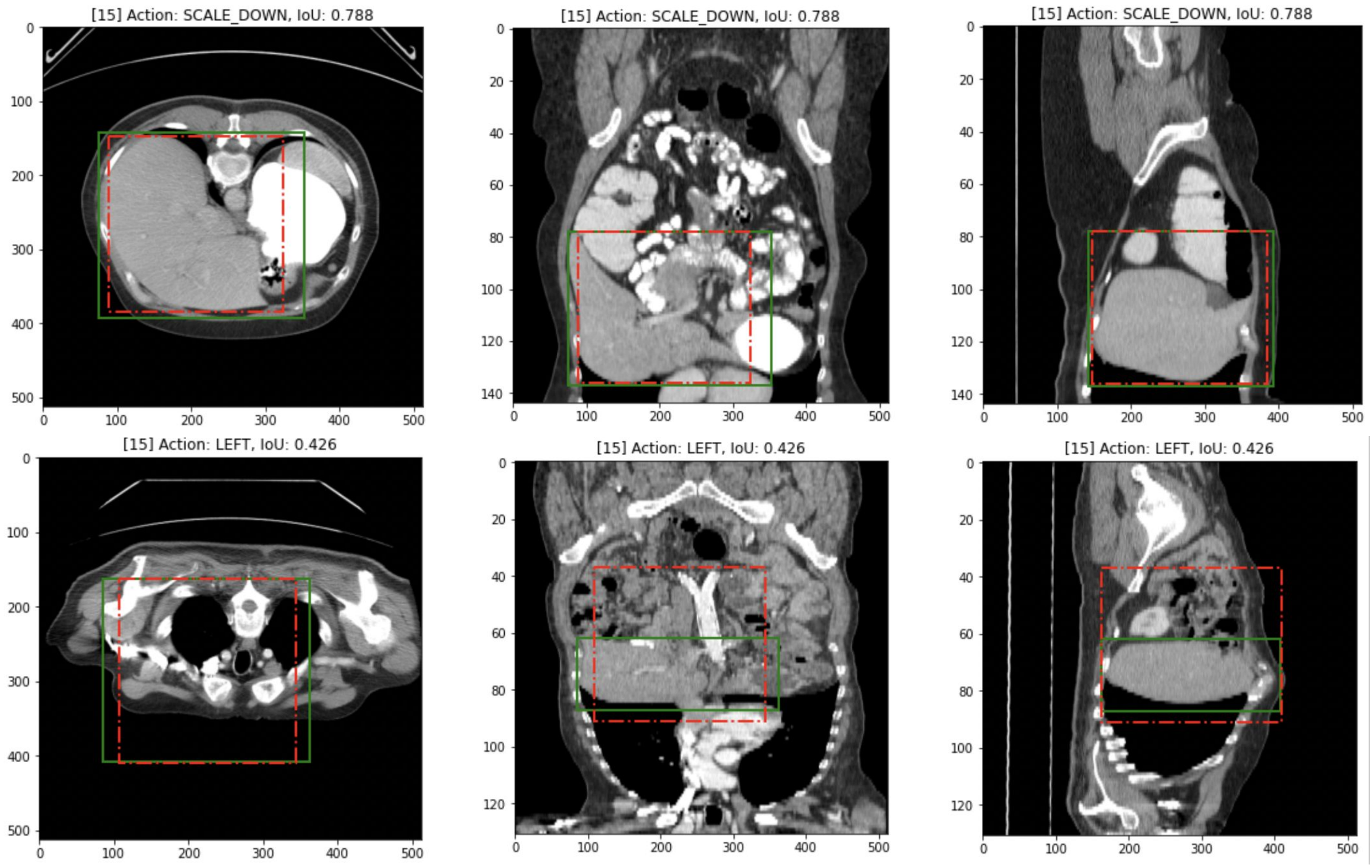
Image 21

IoU Score: **0.79**

Worst Example

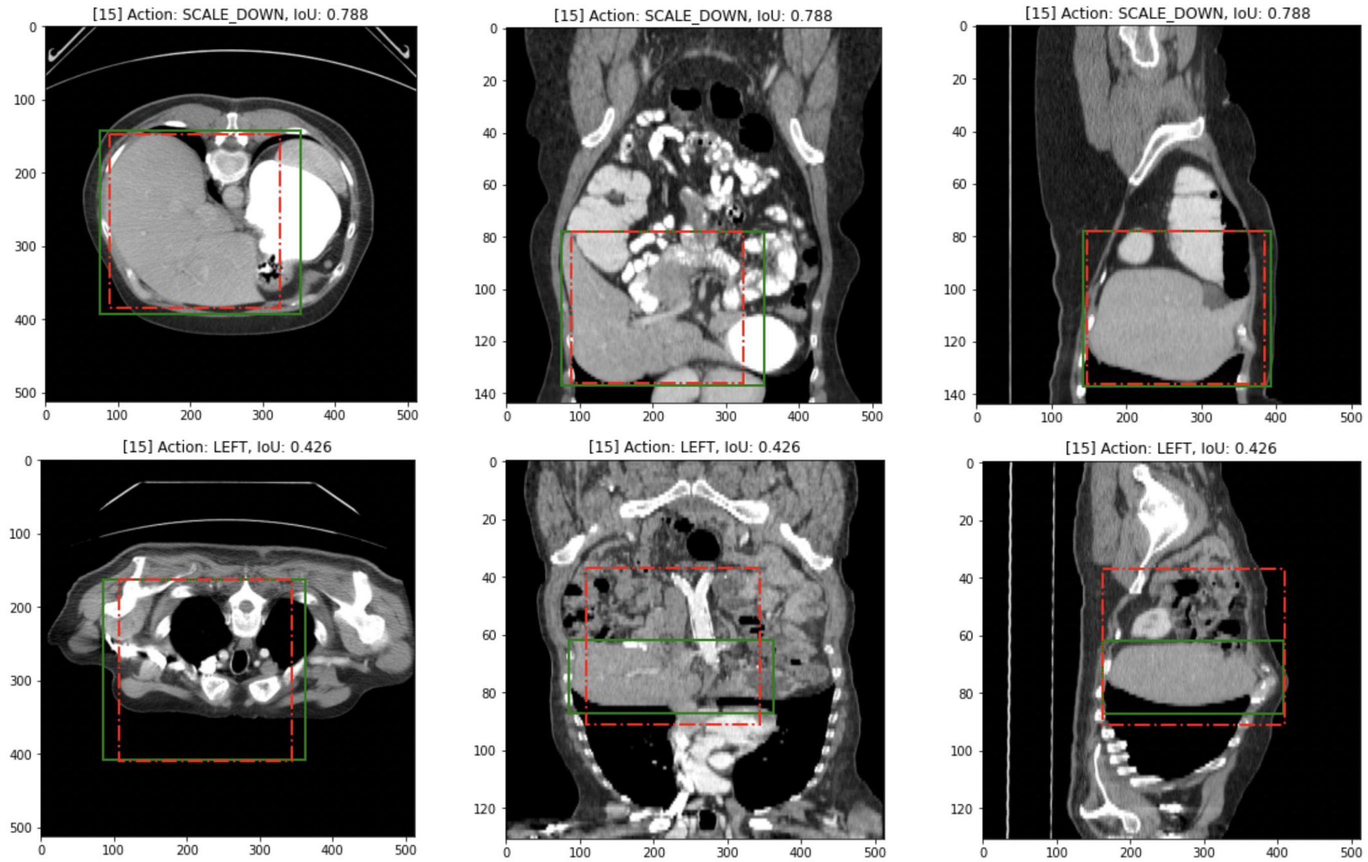
Image 15

IoU Score: **0.426**

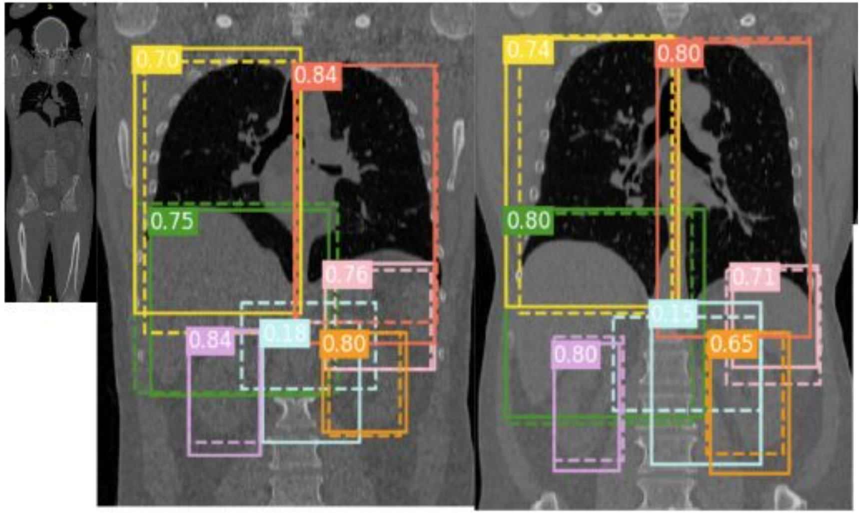


a)

b)


[15] Action: SCALE_DOWN, IoU: 0.788
[15] Action: SCALE_DOWN, IoU: 0.788
[15] Action: SCALE_DOWN, IoU: 0.788
[15] Action: LEFT, IoU: 0.426
[15] Action: LEFT, IoU: 0.426
[15] Action: LEFT, IoU: 0.426

Baseline Model Performance - Original Paper



	Avg IoU	Wall dist [mm]	Centroid dist [mm]
Right Lung	0.77	3.46 ± 5.28	6.06 ± 10.25
Left Lung	0.73	4.91 ± 7.38	10.32 ± 17.09
Right Kidney	0.60	2.96 ± 2.91	5.69 ± 5.67
Left Kidney	0.57	4.06 ± 4.98	7.52 ± 9.02
Liver	0.80	2.41 ± 0.70	3.36 ± 1.34
Spleen	0.60	5.25 ± 7.23	9.20 ± 12.03
Pancreas	0.32	12.26 ± 13.60	20.79 ± 20.38
Global	0.63	5.04 ± 6.01	8.99 ± 10.82
Median	0.60	2.25	3.65

Each organ' truth location and network prediction is illustrated with a different color. Liver ●, right lung ●, left lung ●, right kidney ●, left kidney ●, spleen ●, and pancreas ●. IoU is displayed for every organ. (Visualized better in color)

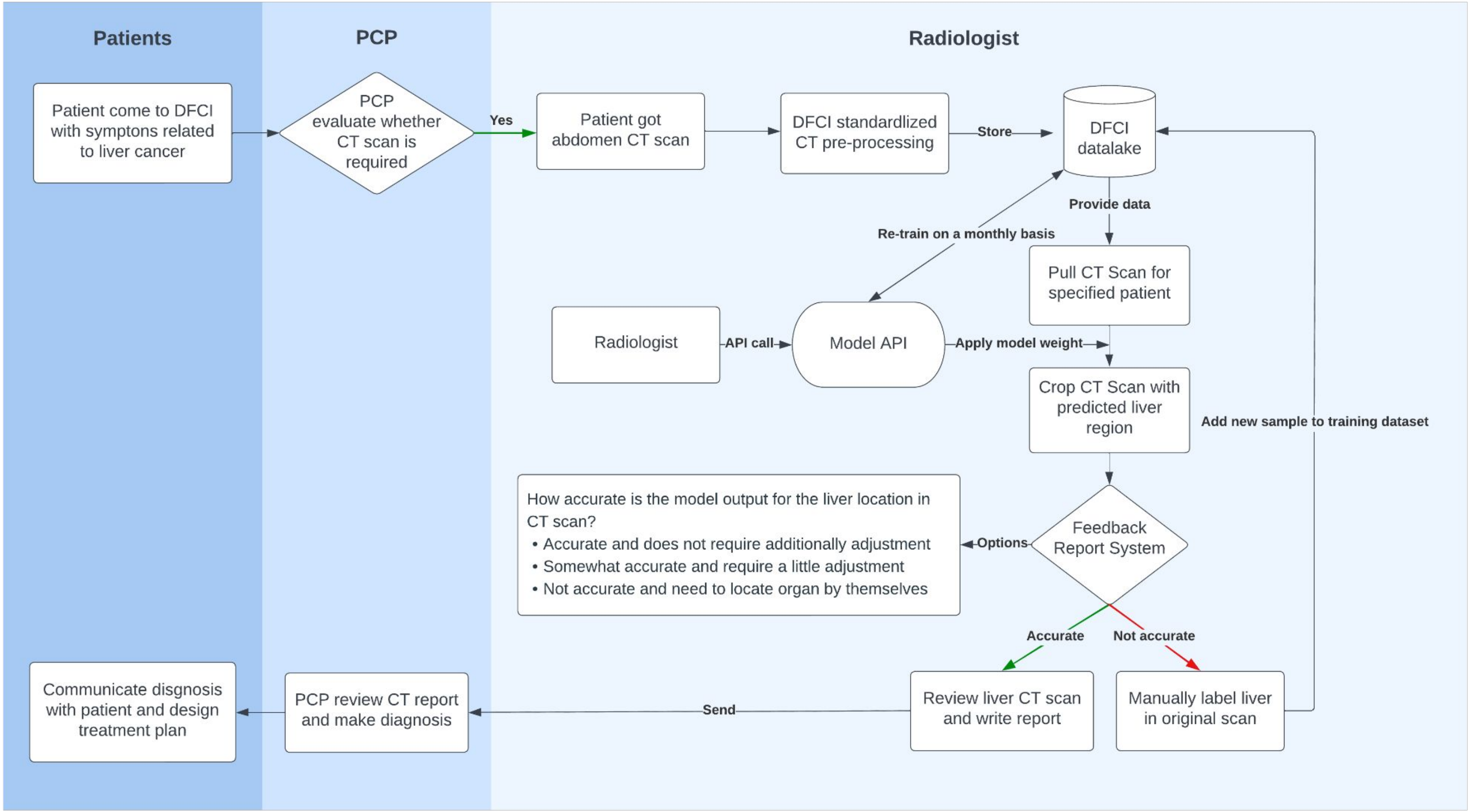
Limitation & Future Direction

Limitation

- Limited training epoch and hyperparameter tuning due to restricted computation resource
- One limitation would be **the generalizability of our model performance in other datasets with varying CT scan volume and resolution.** Even though our Dataset does contain some variability in terms of volume and resolution, it clearly doesn’t cover all possible ranges and therefore may not generalize to other Datasets.

Future Direction

- Train the model on more organ localization tasks using larger datasets to test for its generalizability.
- If the model can achieve high organ localization across different organs on large datasets, it may be possible to deploy this model in hospitals as a part of the CT scan pre-processing pipeline.


Patients
PCP
Radiologist
Patient come to DFCI with symptons related to liver cancer
PCP evaluate whether CT scan is required
Yes
Patient got abdomen CT scan
DFCI standardlized CT pre-processing
Store
DFCI datalake
Provide data
Re-train on a monthly basis
Pull CT Scan for specified patient
Radiologist
API call
Model API
Apply model weight
Crop CT Scan with predicted liver region
Add new sample to training dataset
How accurate is the model output for the liver location in CT scan?
• Accurate and does not require additionally adjustment
• Somewhat accurate and require a little adjustment
• Not accurate and need to locate organ by themselves
Options
Feedback Report System
Accurate
Not accurate
Communicate disgnosis with patient and design treatment plan
PCP review CT report and make diagnosis
Send
Review liver CT scan and write report
Manually label liver in original scan