Liver Cancer Segmentation

Applied Data Science 2022

Róbert Leó Þormar Jónsson
Jannis-Alexander Arnold

December 10, 2022

Contents

1 Introduction

Liver cancer is among the most common types of cancer in the world, with more than 800.000 people being diagnosed annually, ranking it in sixth place among the most frequently observed cancers.[10]. This number grows even more concerning when taking the death rate into account: With a 5-year relative survival rate of 20%, liver cancer is one of the deadliest cancer types.[7] This fact can be observed on a global scale, as primary liver cancer is responsible for the second most cancer-related deaths worldwide.

Treatment options for liver cancer are plenty, ranging from liver transplantation to chemotherapy.[8] These treatment options find success to varying degrees, depending largely on the stage of the disease. According to the American Cancer Society, the 5-year survival rate for individuals diagnosed at an early stage is 35%, while individuals who receive their diagnosis after the tumor has spread to surrounding tissue possess a 5-year survival rate of 12%. [6] It is therefore of critical importance to identify growing tumors as early as possible.

In addition to the threat posed by tumors originating in the liver, a multitude of other types of cancer can metastasize to the liver, making the organ one of the primary cancer sites in the body.[1]

Considering these facts, it is of no surprise that the liver is often analyzed to locate tumors, lesions, and anomalies. Computerized Tomography (CT) scans are a commonly used imaging tool for this purpose. These scans consist of a stack of cross-sectional x-rays, allowing the responsible doctor to analyze the patient's organs diligently in three dimensions.

However, this process is not only expensive but also time-consuming and arduous, as the doctor is required to look at each cross-sectional x-ray individually in order to identify all tumors.

We believe artificial intelligence (AI) has the potential to be of great use to doctors in this regard, aiding them in the detection of liver cancer. We are of the opinion that by employing deep learning methods, the time required by the detection process can be drastically reduced, thereby diminishing the monetary cost to patients and enabling doctors to help more people.

The goal of this project is to train a deep learning model to be able to identify the liver and potential tumors given a CT scan. The model's performance will be compared to the results of the Liver Tumor Segmentation Benchmark (LiTS), a competition with the same goal and dataset as this project. However, this comparison must be taken with a grain of salt, as will be explained later.

All code for this project is accessible on Github at `https://github.com/Robertleoj/APDS-final-project`.

2 Dataset

The dataset we will be working with is from the LiTS. [1] It contains 131 hand-segmented CT scans of the torso, where each voxel is classified as either background, healthy liver, or tumor lesion. The scans are collected from seven different hospitals around the world.

The height and width of each scan is 512 pixels, but the depth varies, ranging from 42 to 1024 pixels. Figure 1 shows a few example scan slices.

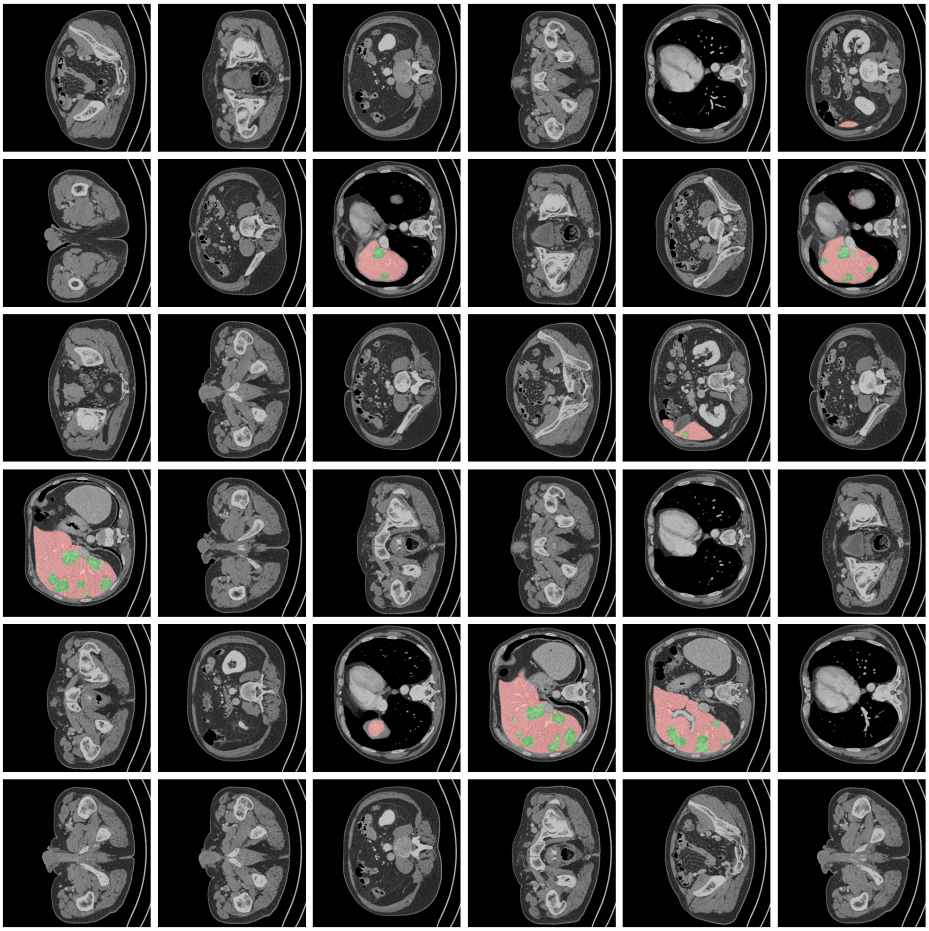

Figure 1: Sample slices

We do not have access to the real test set used in LiTS, so we need to reserve a portion of the data for use as a test set. To this end, we sample 20 scans uniformly from the data. For the validation set, we reserve 10 samples, again sampled uniformly. This leaves us with 101 training samples.

3 Data analysis

When investigating the data further, we discover some interesting insights into the data.

As one is able to observe in Figure 2, the scan depth has a large spread and is right-skewed, meaning that although depth covers a wide range of values, the majority of scans possess little depth. This is also reflected in the interquartile range of the depth, with 190 being the 25th percentile, while 684 is the 75th percentile.

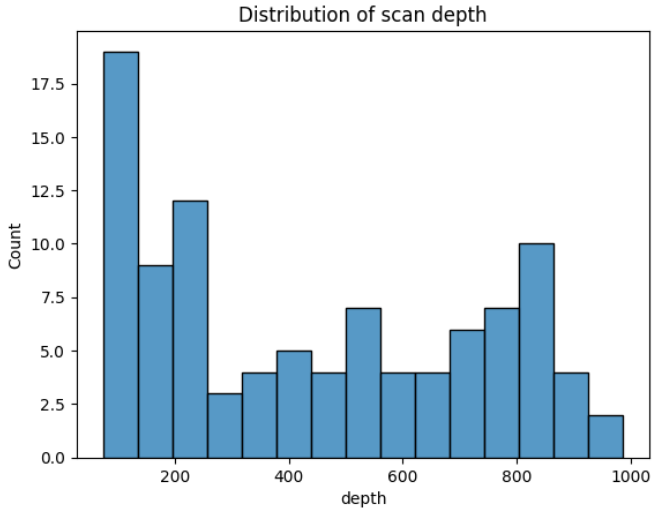

Figure 2: Ratio of scan depths

We also investigate the relationship between livers and tumors in the scans. Figure 3 displays the distribution of liver and tumor size, expressed as the percentage of the scan taken up by liver and tumors, respectively.

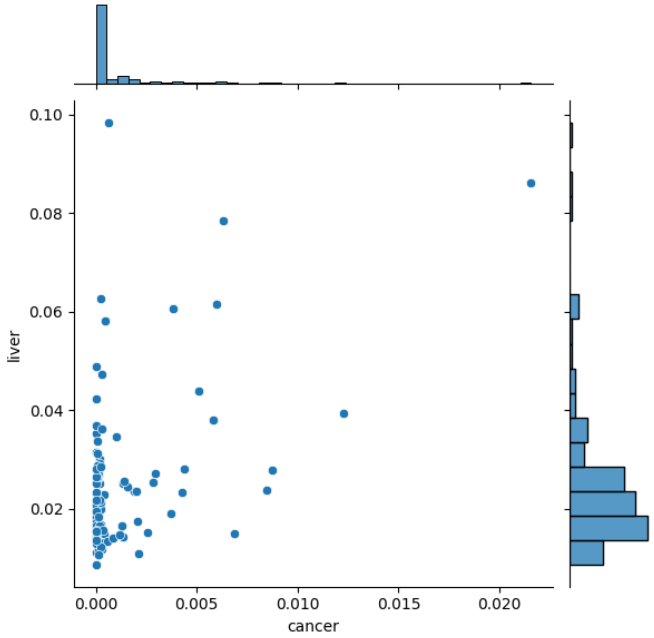

Figure 3: Ratio of liver against ratio of tumor

We see that most scans have only a very small proportion of voxels classified as tumors compared to the proportion of voxels identified as liver.

It is interesting to note is that there seems to be a slight correlation between liver size and cancer size.

Furthermore, it is important to mention that the scans differ in the range of the torso they display. The relationship of the liver's beginning and end can be observed in figure 4.

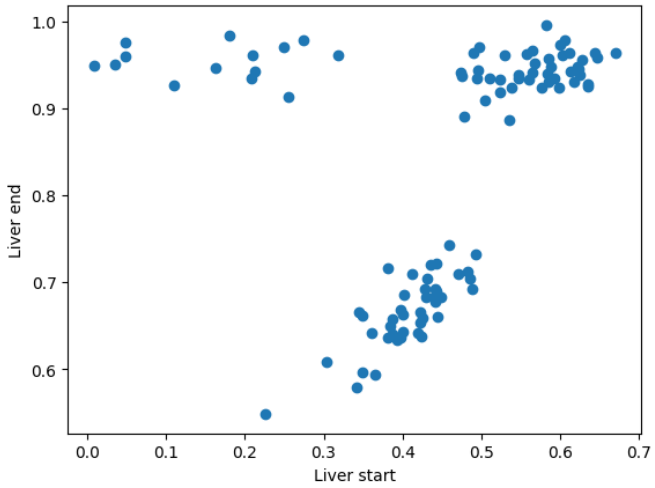

Figure 4: Liver start and end as a ratio of scan depth

This makes it impossible to crop out the liver region as a preprocessing step.

Finally, we take a look at the number of tumors in each scan, measured as the number of connected components classified as tumors. The distribution is shown in Figure 5. The graph is heavily right-skewed, meaning a majority of the scans only have a small number of tumors. This notion is further supported by the interquartile range, seeing as scans containing one tumor comprise the 25th percentile and scans with nine tumors make up the 75th percentile.

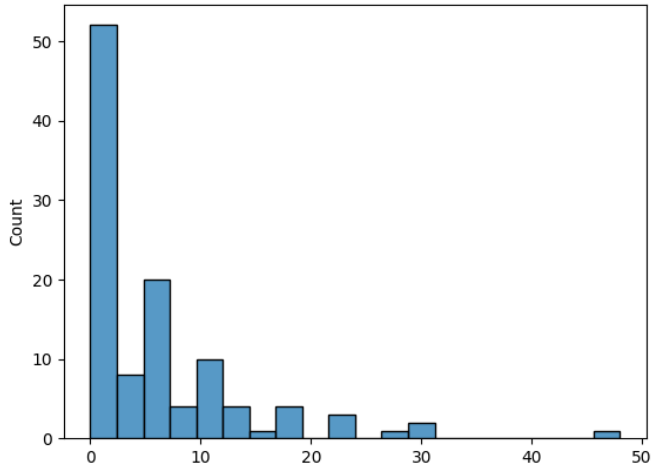

Figure 5: Distribution of the number of tumors in each scan

An interesting bit of information is that the data contains one obvious outlier with respect to the number of tumors, as one scan was categorized as having 47 tumors.

4 Method

In this section, we describe our methodology employed to solve the task.

4.1 General approach

We take a 2.5D approach in order to accomplish the task. Other options, namely a two-dimensional and a three-dimensional approach, were considered but deemed to be less promising and too laborious to implement, respectively. In the two-dimensional case, we lose essential pieces of 3D information that would be very useful for predicting the segmentation. In the three-dimensional case, there are two main problems: The first is that the scans have wildly differing depths, making it hard to perform 3D convolutions reliably. The second one is that 3D convolutions require large amounts of GPU memory, especially if the model has a considerable size. The 2.5D approach retains some 3D information without the issues of using 3D convolutions.

The 2.5D approach specifies the input to the model to be a fixed number of adjacent 2D volume slices stacked in the channel dimension. The model produces an output in the form of the predicted segmentation of the middle slice. This means that the model will be able to rely on information from three dimensions while predicting the segmentation mask of a two-dimensional slice.

The approach is depicted in Figure 6.

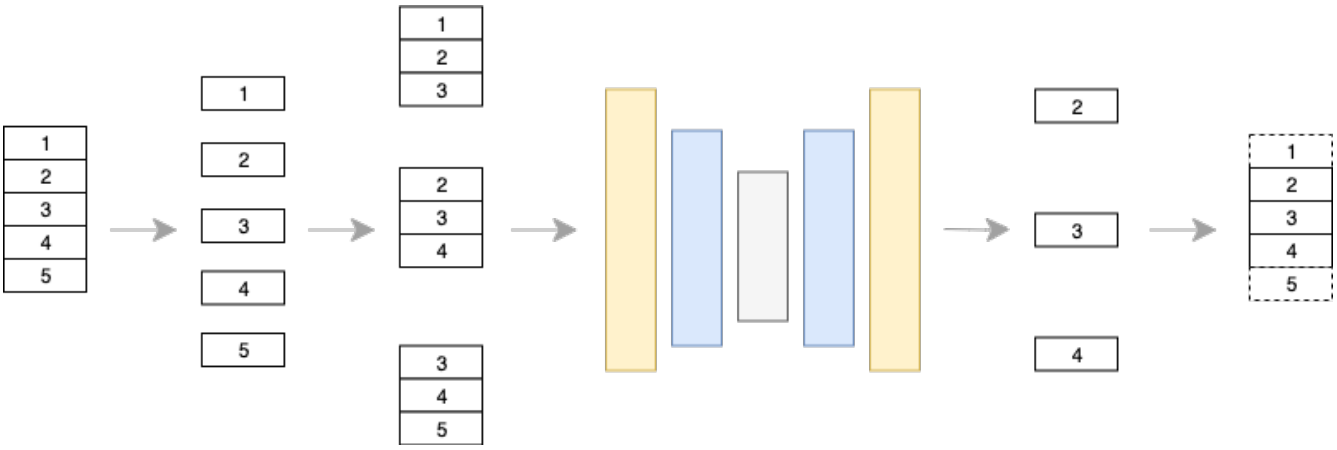

Figure 6: General Approach

For this project, we choose to use five slices as inputs and employ a UNet as our model.

In order to predict the segmentation for a scan, we need to predict the segmentation mask for all the slices belonging to a particular scan, and concatenate them together in the right order. Since we cannot predict the segmentation of the first and last two slices, as they are neighbored by less than two slices on one of their sides, we treat these slices as containing neither liver nor cancer.

4.2 Model architecture

The model we use is based on the UNet architecture. This architectural style is characterized by its two stages: downsampling, in which an encoder is employed to create a semantic representation of the input, and upsampling, in which a decoder uses the semantic representation to create the desired output. Both the encoder and decoder are comprised of a sequence of convolutional blocks. During downsampling, the encoder is responsible for decreasing the spatial dimension and increasing the number of channels between each block, while the decoder does the opposite during upsampling.

For the convolutional blocks, we use two sequential modified ResNet blocks [3]. Our modification is the addition of a Convolutional Block Attention Module (CBAM) to each ResNet block [12]. This module contains a channel attention layer followed by a spatial attention layer, and has been shown to increase performance when added to various convolutional models.

In addition, we insert multi-head scaled linear-time self-attention layers after each convolutional block to allow the network to model long-distance relationships between pixels. [11, 9]. We depict the architecture in Figure 7.

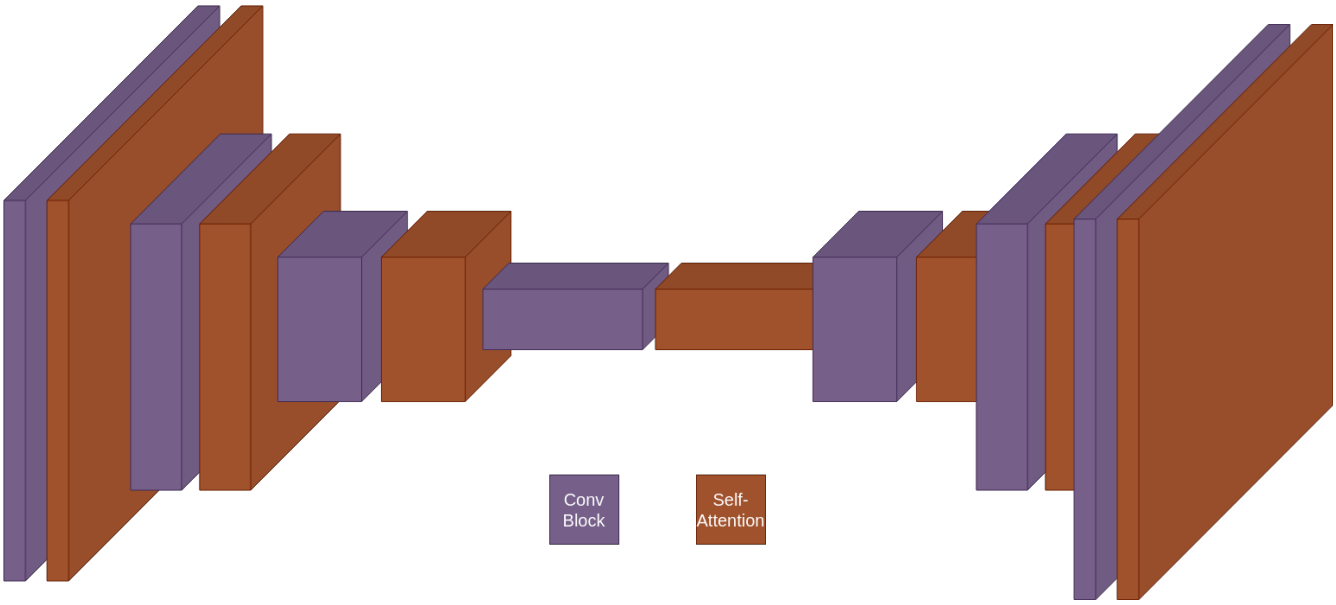

Figure 7: High-level depiction of the UNet architecture

In our model, the encoder downsamples five times, each time halving the spatial dimension, and doubling the channel dimension. The decoder mirrors this procedure, performing the steps in reverse order.

4.3 Preprocessing

We now describe the preprocessing steps performed on the data before feeding it to the model.

The first step is to reorient the scan, ensuring that all scans are oriented the same way. The orientation information required to accomplish this is contained in the scan header.

Next, we clip the HU intensities in the scan to the range $(-200, 200)$. This is done because soft tissue rarely registers outside this range, allowing us to rid the scans of superfluous information.

Since $(-200, 200)$ is not an appropriate range for neural network inputs, we rescale the volume to values between zero and one.

Finally, we slice the volume and mask into 2D slices and cache all of the slices separately. We require a caching system as the preprocessing steps are computationally expensive and it would be inefficient to load the entire scan into memory only to extract a small part of it.

To load a training sample, we sample a scan index i and a slice index j. Then we obtain the four slices surrounding slice j and stack them around it in the channel dimension. The target is slice j of the mask of scan i.

4.4 Postprocessing

We observe that the combined mask of liver and tumors should form a single connected component in a scan. Thus a sensible postprocessing step is to remove all components but the largest one in the predicted segmentation of a scan. This was applied to all outputs of the best model, which we later show to enhance performance.

5 Results

We now present the results of our experiments.

5.1 Performance measure

The performance measure we use is the Dice score. Given two binary masks A and B, it is defined as

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN} \tag{1}$$

We look at the dice scores for the liver and the tumor segmentations separately. To evaluate a model, we average the Dice score for all scans in the validation/test set.

5.2 Final scores

We trained our model for $88,000$ iterations. Using early stopping with respect to the validation set, we found that the model performed best at iteration $75,300$. A plot of the training process is shown in Figure 8.

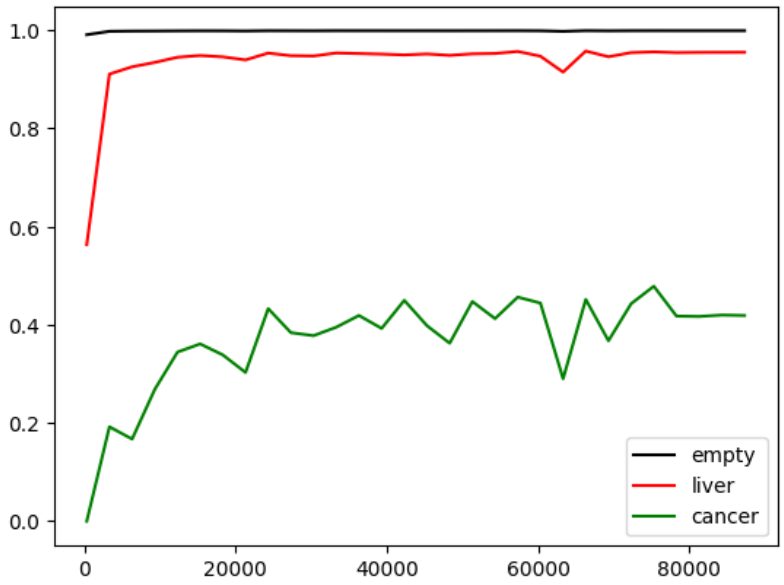

Figure 8: Validation Dice scores during training

Figure 9 shows the scores of the model on the validation set, with and without postprocessing.

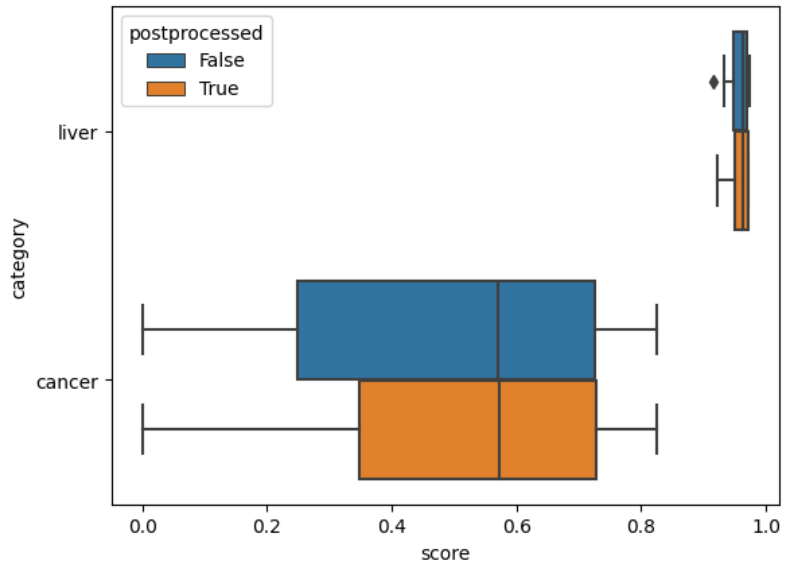

Figure 9: Validation results with and without postprocessing

We see that the results are better when the postprocessing is applied. The mean Dice score for the cancer segmentation was 0.491, while the liver segmentation received a score of 0.957. The scores on the test set were considerably lower. Figure 10 shows the distribution of scores on the test set.

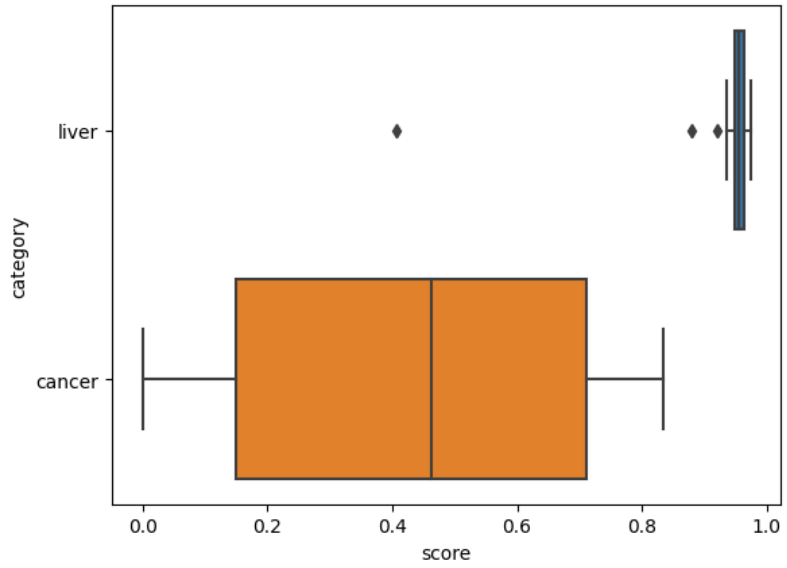

Figure 10: Test set scores

The mean Dice scores for tumor and liver segmentations were 0.437 and 0.925, respectively.

This places us at roughly the 30th percentile in all three LiTS competitions. However, there are three reasons why it might be misleading to compare our scores to the results achieved in the competitions.

The first is that their test set is different from ours. As can be seen from the box plots, the spread of the Dice scores for tumor segmentation is very large, meaning that the composition of the test set can greatly affect the outcome.

The second reason is that in the competition, competitors could train on all 131 scans, as the test set was externally held. Clearly, this can greatly increase performance.

The third and final reason is that the competitors were allowed to see their scores on the test set three times each day over the span of a week. This means that competitors could optimize their model to obtain higher test scores.

Irregardless of how our scores compare to the competition's scores, we consider our experiment to be a success, as we placed well without receiving the advantages that entering the competition would have granted us.

5.3 Interpretation: Spatial attention visualization

As mentioned above, the CBAM module contains a spatial attention layer. Although the model belongs to the category of black-box models due to its nature as a neural network, it is insightful to observe the spatial attention masks that the model learned. However, it is not fruitful to observe them at stages where the input is not 512×512 as by that point, the network has transformed the image into a semantic representation that is harder to interpret.

There are four spatial attention modules in the model that have a 512×512 output. The first two can be found in the first convolutional block in the encoder, and the other two are situated in the last block of the decoder. We visualize them for a single slice in Figure 11.

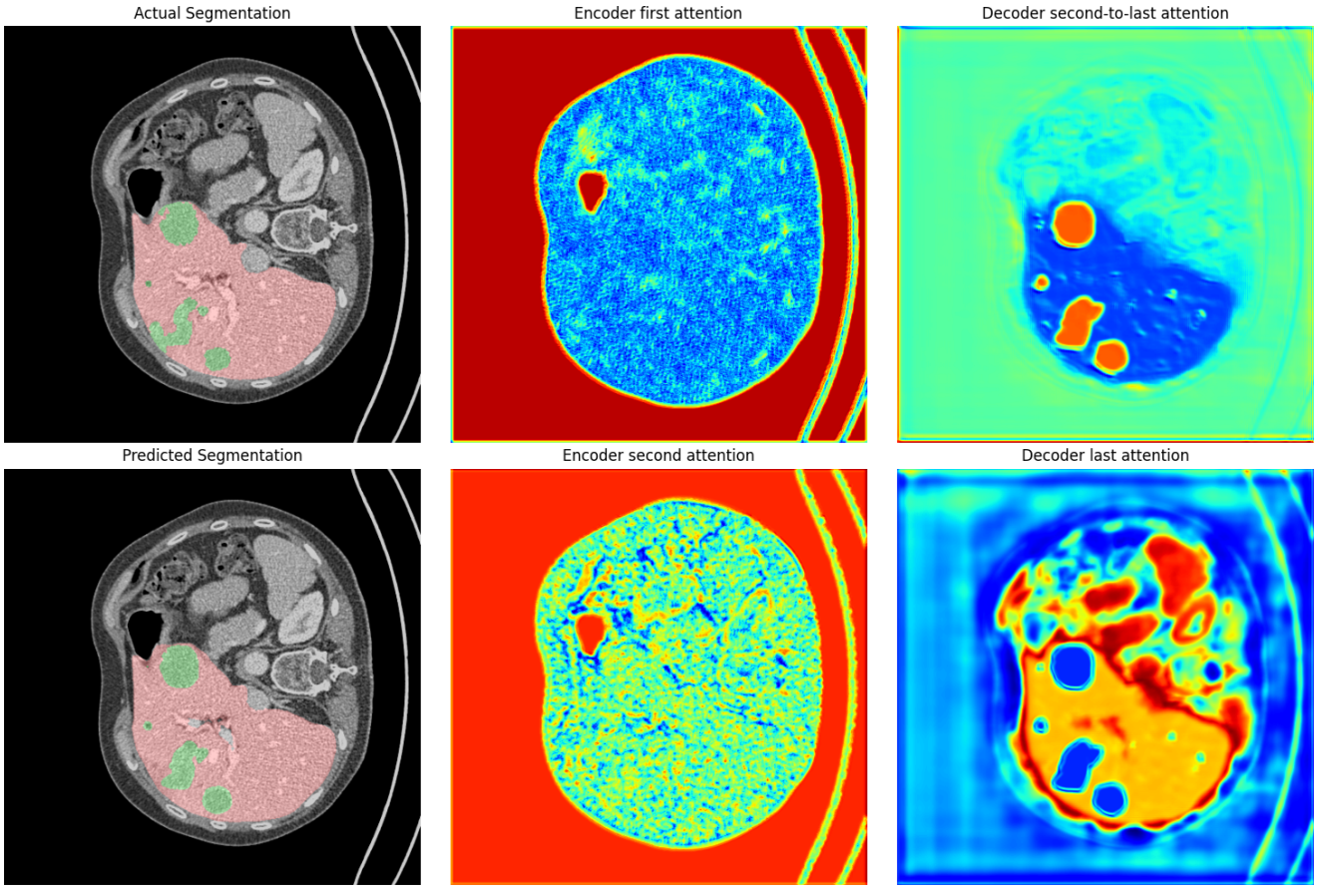

Figure 11: Visualization of spatial attention activations

We observe that both attention modules in the encoder place high attention on low-intensity pixels. It is difficult to interpret

this, as the input to the attention modules has already gone through several convolutional layers. However, one could hypothesize that the model uses this to extract the outline of the body in the image. It could also be the case that the model attempts to identify pixels as background in order to disregard them.

The attention masks in the decoder offer a more insightful picture. The second-to-last mask is clearly focusing on the tumor lesions, as the high-intensity areas of the mask correspond almost exactly to the predicted tumors. It is important to note the mid-intensity outline of the tumors in the mask.

The last mask in the decoder is difficult to interpret. The highest intensity areas are neither liver nor tumors. Instead, areas of high intensity appear to focus on the other organs in the scan. We are not sure how to interpret this. It is interesting to note that the tumors are very low-intensity in this mask, as opposed to its predecessor, where they were marked by high-intensity areas. Unfortunately, the purpose behind this eludes us.

6 Ethical and societal implications

AI first saw deployment in a medical environment in the early 1970s with MYCIN, a program that aided doctors in identifying blood infection treatments. [4] Ever since these humble beginnings, AI's role in the profession has gradually increased, contributing to the medical field in a multitude of ways. Today, AI has become a quintessential part of the medical industry as a tool utilized in various contexts, ranging from radiology to psychiatry. However, there appear to be concerns regarding AI's application in interactions with patients. Surveys such as the one conducted by Khullar D. et al [5] suggest a degree of apprehension towards AI in the public eye. The question inquiring as to whether or not respondents would like an AI with 98% accuracy to diagnose cancer without them being able to comprehend its decision is especially intriguing, as a majority of respondents stated that they felt uncomfortable with the described scenario, even though the rate of misdiagnosis for cancer lies between 10% and 20%. [2] One can therefore infer that one of the reasons behind the distrust toward AI is related to its inherent lack of transparency. From a deontological perspective, humans desire to understand the reasons behind actions before making moral judgments, which is a nigh impossible task with black-box models.

This dilemma becomes apparent in matters concerning life and death. One can hardly imagine the emotional consequences for all parties involved if a doctor had to explain to family members of a deceased patient that their loved one died because the model made an inexplicable prediction. Furthermore, the legal ramifications of such a scenario would be gargantuan. The discussion of the question regarding who is to be held responsible for the death of the patient frankly exceeds the scope of this work but is one that must be held in the context of the application of AI in treatment prescription.

Focusing the discussion on this project's use case, it is of great importance to talk about the biases present in the data.
The first type of bias that needs to be mentioned is response bias. Due to the fact that CT scans are primarily taken when symptoms are experienced by the patient, a heavy bias towards cancer exists in the data. This prompts the model to predict cancer where none is present.
Another type of bias that is relevant to discuss is the potential for system drift. It is within the realm of possibility that a shift to the system occurs. For example, a change in CT scan technology or a newly discovered type of cancer might lead the model to make biased predictions.
Additionally, it is important to mention how outliers are treated by the model. Since the model is trained on CT scans depicting people who mostly conform to the norm, it will have a hard time making accurate predictions for people deviating from normalcy. We observed this on one of the scans belonging to the test set, where a Dice score of 0.4 on the liver segmentation was obtained. This was due to the fact that the patient's organ composition was outside the norm.

Considering these points, we must stress that the model is only to be used as a reference point and should never be employed as a diagnostic agent. We envision this technology's ideal use to serve doctors in recognizing areas of interest in the liver which can then be further investigated by them. We recommend that doctors further educate themselves in the area of AI to not only understand the model's outlines but also its capabilities and deficiencies.

References

[1] Patrick Bilic et al. “The liver tumor segmentation benchmark (lits)”. In: *Medical Image Analysis* (2022), p. 102680.

[2] *Cancer statistics - facts on cancer*. URL: https://paulandperkins.com/cancer-statistics/.

[3] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[4] XSOLIS Insights. *The evolution of AI in Healthcare*. URL: https://www.xsolis.com/blog/the-evolution-of-ai-in-healthcare.

[5] Dhruv Khullar et al. “Perspectives of Patients About Artificial Intelligence in Health Care”. In: *JAMA Network Open* 5.5 (May 2022), e2210309–e2210309. ISSN: 2574-3805. DOI: 10.1001/jamanetworkopen.2022.10309. eprint: https://jamanetwork.com/journals/jamanetworkopen/articlepdf/2791851/khullar_2022_ld_220084_1650983640.36472.pdf. URL: https://doi.org/10.1001/jamanetworkopen.2022.10309.

[6] *Liver cancer - statistics*. Oct. 2022. URL: https://www.cancer.net/cancer-types/liver-cancer/statistics#:~:text=Survival%5C%20rates%5C%20depend%5C%20on%5C%20several%5C%20factors%5C%2C%5C%20including%5C%20the%5C%20stage%5C%20of,year%5C%20survival%5C%20rate%5C%20is%5C%2012%5C%25..

[7] *Liver cancer survival rates: Cancer of the liver survival rates*. URL: https://www.cancer.org/cancer/liver-cancer/detection-diagnosis-staging/survival-rates.html.

[8] *Liver cancer treatment*. URL: https://www.cancer.gov/types/liver/what-is-liver-cancer/treatment.

[9] Zhuoran Shen et al. “Factorized Attention: Self-Attention with Linear Complexities”. In: *CoRR* abs/1812.01243 (2018). arXiv: 1812.01243. URL: http://arxiv.org/abs/1812.01243.

[10] medical The American Cancer Society and editorial team. *Key statistics about liver cancer*. Jan. 2022. URL: https://www.cancer.org/cancer/liver-cancer/about/what-is-key-statistics.html.

[11] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[12] Sanghyun Woo et al. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.