Medical Imaging - Coursework

William Knottenbelt, wdk24

Word Count: 2869

Introduction

In this report we trained a small version of U-Net [1] for lung segmentation in CT scans, using 12 cases from the Lung CT segmentation challenge (2017) dataset [2] (8 for training, 4 for testing). We assessed performance of the model quantitatively using metrics like binary accuracy and the dice similarity coefficient, and visually inspected many predictions against the ground truth for qualitative judgement. These performance measures are plotted in various forms, and the results are discussed.

(a) Methodology

Preprocessing

Each case in the LCTSC dataset consists of multiple individual CT scans, stored in separate DICOM files, which collectively form a 3D volume of the patients body. According to the DICOM metadata, the dataset is comprised of 6 males and 6 females, and most of the measurement parameters are consistent for all cases (scanner manufacture, voltage, patient position relative to equipment etc.). All cases except case 009 also had the same scanner model and exposure. For de-identification purposes, we removed the patient's name and birth date/time from the DICOM's where necessary (cases 003, 005, 007), in accordance to the PS3.15 Basic Application Level Confidentiality Profile [3].

The ground-truth lung segmentations were provided in npz files as 3D numpy arrays per case. After visually inspecting the images and masks, we discovered that the orderings of the DICOM files with respect to the segmentation masks and the patients' body orientation varied for different cases (further details about this can be found in the docstring of the `preprocessing.py` script). To avoid confusion and bugs, we constructed 3D arrays of the images correctly aligned with the segmentation arrays, and saved them in npz files. We also found significant class imbalance, with only 3.84% of the total pixels in the dataset being part of a mask.

U-Net

Our model architecture was a small version of U-Net [1], consisting of an encoder with 3 downsampling blocks and a decoder of 3 upsampling blocks to restore the original dimensionality. The downsampling increases the receptive field for the same number of layers, allowing for the extraction of global features.

Skip connections between the encoder and decoder were used so that the decoder could consider both local features and global features of the image. We also used batch normalization to stabilize training. For segmentation, we apply a sigmoid function to the output logits to obtain probabilities for each pixel belonging to the mask. A threshold of 0.5 is used to convert the probabilities to a binary mask.

Loss function

There are many possible loss functions capable of training semantic segmentation models [4]. One such example is binary cross entropy (BCE) loss, which minimizes the negative log likelihood of the ground truth pixels. It is an extremely widely-used loss function for binary classification tasks, favoured for its probabilistic interpretation and its ability to produce continuous, smooth gradients for stable training. However, BCE treats each pixel equally and independently, hence for imbalanced datasets where the masks are small compared to the image size, BCE may encourage the model to be biased towards the majority class, as it dominates the loss calculation.

Soft Dice loss [5], inspired by the Dice Similarity Coefficient (DSC), solves this issue by only focusing on the agreement of the predictions with the ground truth mask, regardless of their size relative to the image. It is defined:

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} t_i \cdot p_i + s}{\sum_{i=1}^{N} t_i + \sum_{i=1}^{N} p_i + s}, \tag{1}$$

where N is the number of pixels, t_i is the ground truth label of pixel i, p_i is the predicted probability of pixel i being part of the mask, and $s = 1$ is the smoothing term. The smoothing term prevents division by zero, allows for a non-zero gradient in the case of an empty ground-truth mask, and generally smooths the loss surface for more stable training. Despite this smoothing term, dice loss is often susceptible to unstable training since it can be sensitive to small changes in predictions (particularly when the predicted mask is small), leading to an uneven loss surface.

To leverage the robustness of Dice loss to class imbalance and the stable training of cross-entropy, Taghanaki et al. (2019) [6] proposed Combo Loss, which is defined:

$$L_{Combo} = \alpha \cdot L_{m\_BCE} - (1-\alpha) \cdot \frac{2\sum_{i=1}^{N} t_i \cdot p_i + s}{\sum_{i=1}^{N} t_i + \sum_{i=1}^{N} p_i + s}, \tag{2}$$

where $\alpha \in [0, 1]$ controls the contribution of BCE, and L_{m_BCE} is the modified cross entropy loss, given by:

$$L_{m\_BCE} = -\frac{1}{N}\sum_{i=1}^{N} \left[\beta \cdot t_i \cdot \log(p_i) + (1-\beta) \cdot (1-t_i) \cdot \log(1-p_i)\right], \tag{3}$$

where $\beta \in [0, 1]$ allows us to control the penalization for false positives/negatives ($\beta < 0.5$ penalizes false positives more). For this project, we used Combo Loss with $\alpha = 2/3$ and $\beta = 0.5$. This corresponds to an equal contribution of regular BCE and soft dice loss, since $\alpha \cdot \beta = 1/3 = (1-\alpha)$.

Train-Test split

We chose to split the dataset into 2/3 train and 1/3 test, as this provides sufficient examples to train the model well, while having a large test set to do a rigorous evaluation of the model. However, it is necessary to split the data by case (patient) rather than splitting all the images randomly since nearby images of the same patient are highly correlated as they are images of the same body. Hence images from the same case in the train and test set would cause data leakage, and we would not have an independent test set to provide an unbiased assessment of the model. We used cases 2,5,10,11 for testing and all others for training. This split was selected to minimise domain shift since it yielded a near-identical proportion of empty masks to non-empty masks (0.43696 in train set, 0.43643 in test set), and both sets consisted of half females and half males. This split yielded 1158 training cases and 527 test cases (approximately 2/3 train and 1/3 test).

Training

We trained the U-Net for 10 epochs using stochastic gradient descent with a learning rate of 0.1 and a batch size of 3. After each epoch, we evaluated the model on the full test set and logged the loss, mean accuracy and mean dice similarity coefficient for both the train and test set predictions. This was done so that we could visualise the model performance during the training phase to ensure it was learning effectively and to detect over-fitting.

Performance Metrics

We used a number of metrics to quantitatively assess the model predictions. One such metric is the Dice Similarity Coefficient (DSC), which is defined:

$$DSC = \frac{2TP}{2TP + FP + FN}, \tag{4}$$

where we define positives as pixels belonging the mask, and negatives as pixels that do not. Then TP are the true positives (overlap between predicted mask and ground truth), FP are the false positives and FN are the false negatives. DSC ranges from 0-1, with 0 being no overlap and 1 being a perfect prediction. In the edge-case of an empty mask and empty prediction, we use the convention $DSC = 0$ so that there is not a radical difference between the prediction of a couple of mask pixels vs an empty prediction when there is no ground truth mask. This convention means that the DSC is always zero for an empty ground truth mask, making it meaningless in this case.

We used several other metrics for evaluating the model, as listed below:

- Binary Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{5}$$

In the case of small ground-truth masks relative to the background, this metric can be deceptive, giving high values for low-quality predicted masks as long as they are small (since TN is high).

- Recall (Proportion of positives correctly classified)

$$Recall = \frac{TP}{TP + FN}. \tag{6}$$

- Specificity (Proportion of negatives correctly classified)

$$Specificity = \frac{TN}{TN + FP}. \tag{7}$$

- Precision

$$Precision = \frac{TP}{TP + FP}. \tag{8}$$

(b) Results & Analysis

In Fig. 1, we plot the evolution of three metrics over the training process. It should be noted that a single epoch consisted of 386 training steps, meaning that the model likely improved substantially over the first epoch. Hence, it is natural that the test-set metrics start off better than the train metrics, as the test-set evaluation occurs at the end of each epoch. Fig. 1(a) shows the combo loss for the train and test sets. We see that the training loss decreases sharply during the early stages, then levels out after around three epochs. The test loss is much flatter than the train loss, decreasing less over the training process, but nevertheless decreasing consistently. This disparity between test and train loss could imply over-fitting.

Fig. 1(b) plots the mean dice similarity coefficient (DSC) of the subset of the examples which have a non-zero ground truth mask. This is necessary as the DSC is always zero for empty-masks (using our convention mentioned earlier), hence it is meaningless for these examples. We see that the mean DSC is very similar for the train and test sets during training, which indicates good generalisation, and refutes the hypothesis of over-fitting. It is possible that as the training progresses, the model becomes increasingly certain about it's predictions in the training set as it has seen them before. This may push the probabilities to the extremes, causing a disparity between the train/test loss but not the DSC, which depends only in the binary predictions and not the probabilities.

Fig. 1(c) shows the mean accuracy on the train and test sets consistently improving over the whole process. We see that the test accuracy is higher than the train accuracy for all epochs, which may indicate some domain shift between the train and test sets. Since the proportion of positives (mask pixels) in the test set is lower than the train set (0.03402 vs 0.04035), one hypothesis is that the model is biased towards negatives (the majority class) and hence it has higher accuracy in the test set where there are more negatives.

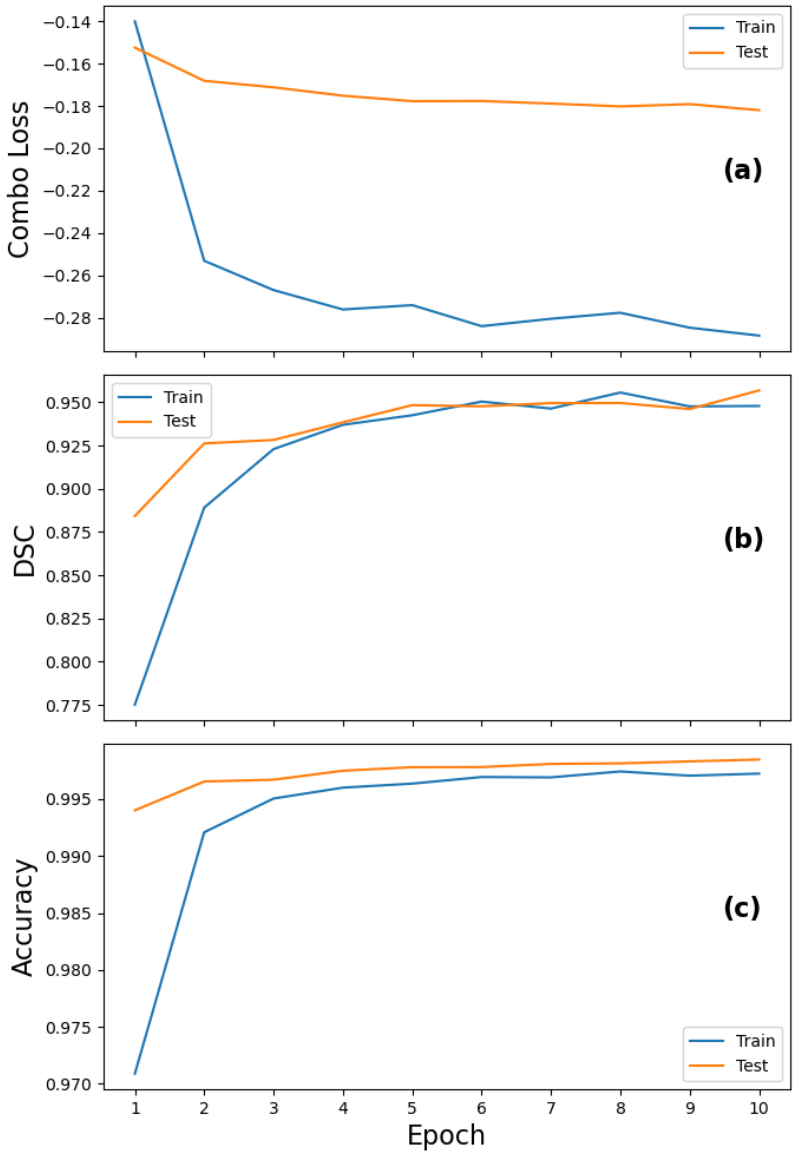

Figure 1: Plot showing evolution over the training process of (a) Combo loss (b) mean DSC for subset of images with non-empty ground truth mask (c) mean accuracy, for both the train and test sets.

After training the model, we made predictions for the entire dataset. The model has a mean accuracy of 0.9986 in the training set, and 0.9985 on the test set. We computed the DSC for the images with non-zero ground truth masks in each set, yielding an average of 0.970 for the training set and 0.958 for the test set. These high scores indicate strong performance, and good generalisation due to the similarity between the train and test sets.

We also calculated the recall and specificity across all pixels in the test set collectively, yielding 0.9737 and 0.9994, respectively. These metrics are both high, indicating good overall performance, however since the specificity is higher than recall, it implies that the model is slightly biased towards predicting negatives over positives, which is likely a result of the overwhelming majority of negative pixels in the dataset. To improve this, we could lower α in equation (2) to increase the contribution of dice term in the combo loss, so that the class imbalance has a smaller effect on the model. We could also increase β to penalize false negatives more, encouraging the model to make more positive predictions.

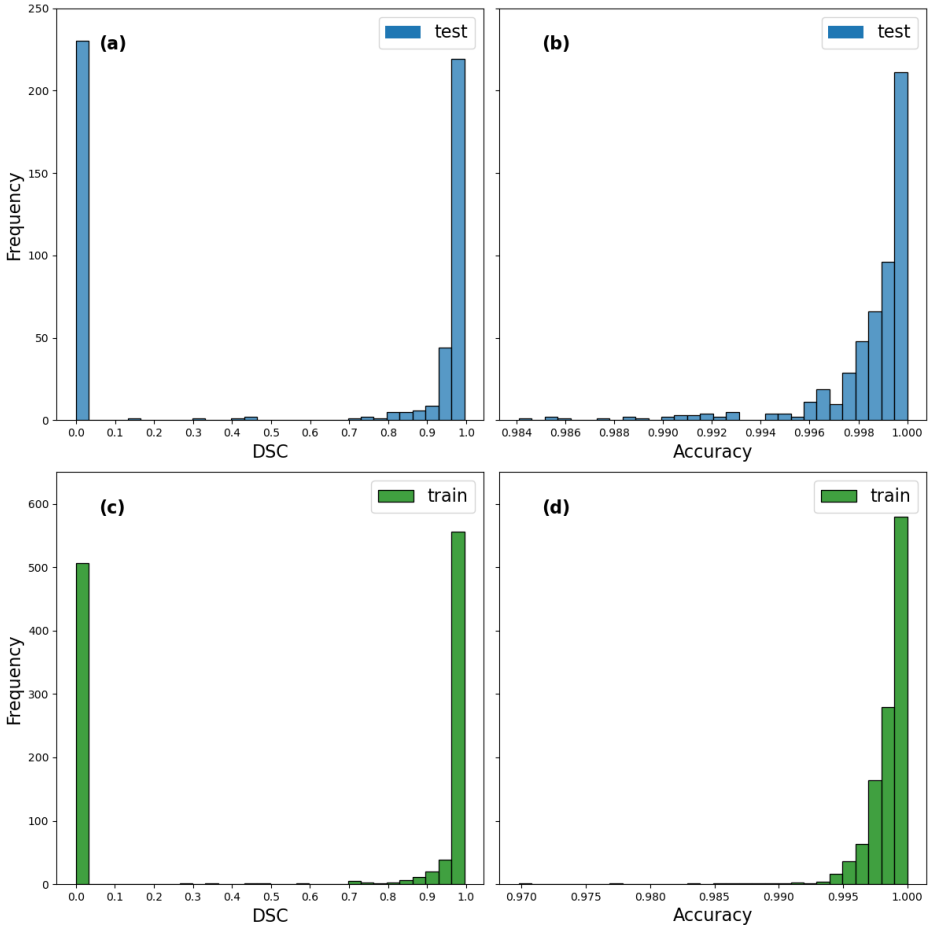

Figure 2: Frequency histograms showing the distribution of (a) test set DSC (b) test set Accuracy (c) training set DSC (d) training set Accuracy.

In Fig. 2 we plot histograms of the DSC and Accuracy for the train and test sets. We can see that they are extremely similar for both sets, suggesting good generalisation. The distribution of DSC is dominated by very high scores (close to 1) and zeros, which is a result of the fact that the DSC is zero for images with empty-masks (roughly 44% of both the training and test sets). The high scores

indicate strong model performance. We also observe that the accuracy is consistently high, with the lowest accuracy of any prediction being 0.97.

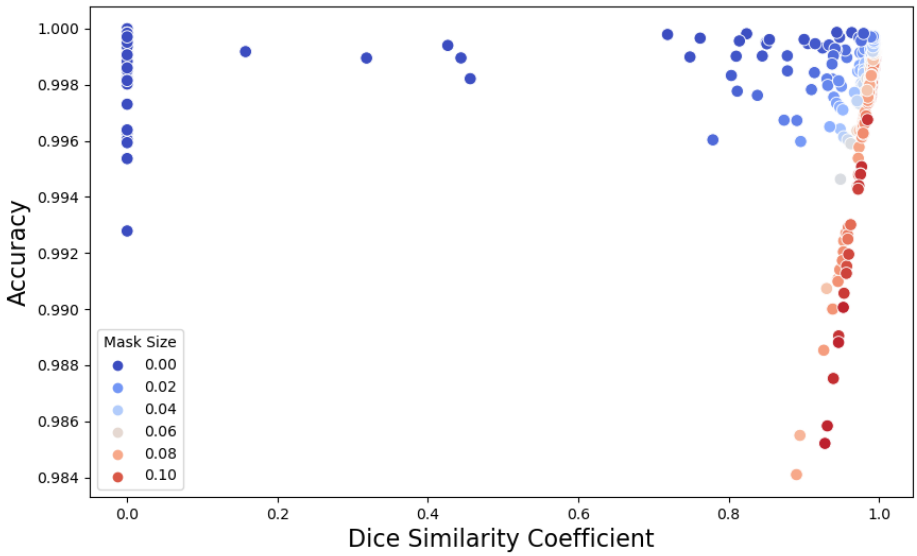

Figure 3: Scatter plot of Accuracy against DSC for images in the test set, with colour indicating the size of the mask relative to the image.

Fig. 3 shows a scatter plot of accuracy against DSC for images in the test set, coloured according to the mask size. We observe a positive correlation between accuracy and DSC for images with mask sizes of more than 0.05 of the image. This is expected as a better model prediction is naturally associated with higher DSC and accuracy. We also observe that the region of higher accuracy and lower DSC is characterised by small masks (below 0.04 of the image). This is showcases why DSC is an important metric. For images where the mask is small compared to the image, the model can achieve high accuracy merely by predicting a small mask, regardless of the masks quality. However, DSC focuses specifically on the overlap between the predicted and true masks, thus a lower quality mask causes a lower DSC.

In Fig. 4 we plot heatmaps of the mask size, DSC and accuracy for all image slices in each case of the test set. The slices are ordered from the torso region of the body (early indices) upwards to the neck/head region (late indices). Unsurprisingly, we observe in Fig. 4(a) that the early and late regions of the body have no mask, since they do not contain lungs. Naturally, we see from Fig. 4(b) that these regions have $DSC = 0$, since this is always the case for empty masks. It is clear that these regions are responsible for the spike of $DSC = 0$ in Fig. 2(a). Slices from the region of the body containing the lungs (middle) have high DSC scores, with the lowest non-zero scores coming from the top and bottom ends of the lungs, where the masks are small. The DSC appears similar across cases, except slightly worse for case 011. In Fig. 4(c) we see that all regions of the body have predictions with high accuracy. The worst accuracy was found in slices of Case 011, which are visualised in Fig. 6.

We visually inspected many images, masks and predictions from the test set over a wide range of DSC and accuracy values. 38.1% of the test set had $DSC > 0.975$ and appeared almost perfect, as shown in Fig. 5. Intermediate DSC scores in the range $(0.8, 0.975]$ constituted 16.5% of the test-set, and is visualised in Fig. 7. We see that, although there are some visual imperfections, the model has

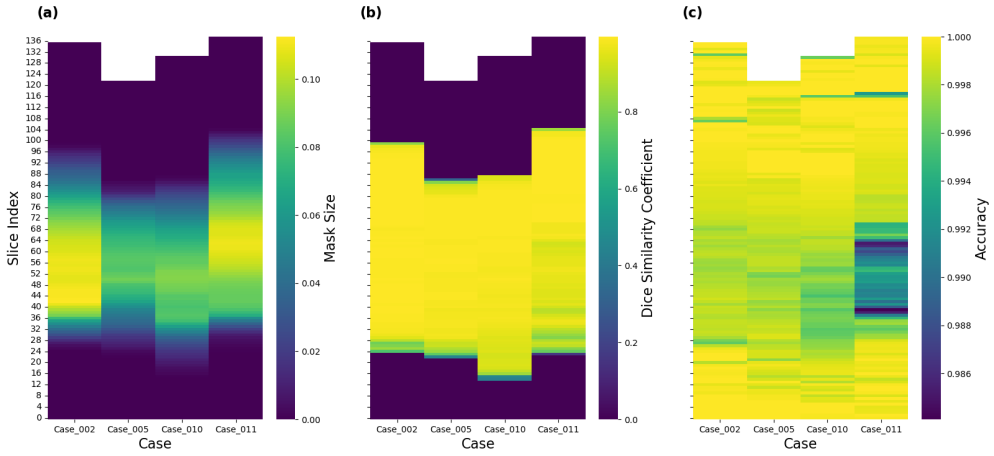

Figure 4: Heatmaps of the image slices in each case (patient) of the test set, with colour representing (a) Mask size relative to image, (b) Dice similarity coefficient of prediction, (c) Accuracy of prediction.

still captured the shape and location of the lungs in each image very well. The slices with intermediate DSC scores exhibited the widest range of accuracy and mask sizes, as can be seen in the scatter plot (Fig. 3), with larger masks associated with lower accuracy.

Examples with non-empty masks and low $DSC < 0.8$ made up just 1.7% of the test-set, and were characterised by small masks and visually inaccurate predictions, as shown in Fig. 8. This is consistent with the heatmap in Fig. 4(b), where we see the lower non-zero DSC scores come from the top and bottom ends of the lungs, where the masks are small but non-zero. 43.6% of the test-set were examples with empty ground-truth masks, out of which 35.6% have perfect accuracy, 59% have accuracy $\in [0.999, 1)$, and 8.3% have accuracy < 0.999. Three of these are visualised in Fig. 9.

Based our visual inspection, we subjectively categorise predictions as 'excellent' if they have $DSC > 0.975$ or an empty ground truth mask with accuracy > 0.999. This makes up 78.2% of the test set.

Finally, to gauge an overall evaluation of the model, we plotted a precision-recall (PR) curve using all probability predictions of pixels in the test set collectively, depicted in Fig. 10. The area-under-the-curve (AUC) measure is an impressive 0.996, indicating exceptional performance. Notably, precision and recall both concentrate on the accuracy of the predicted masks versus true masks, disregarding the background, and thus are not biased by the majority of negative pixels.

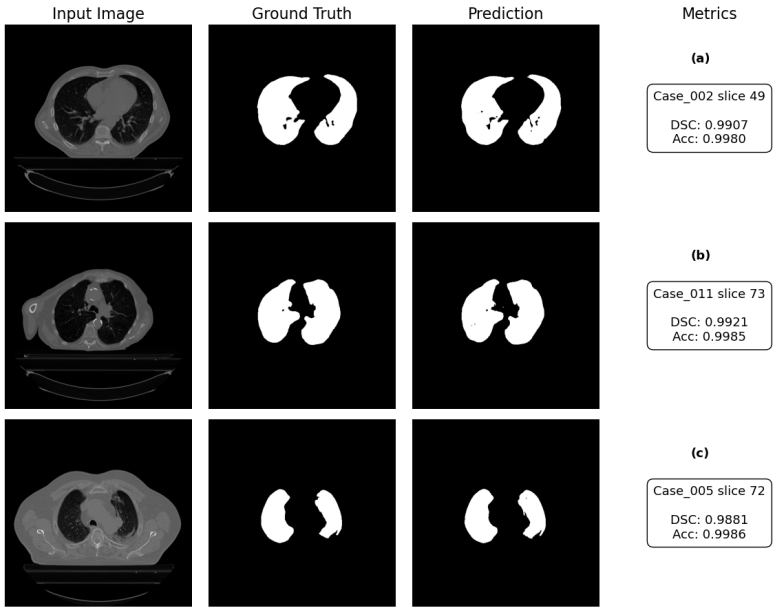

Figure 5: Visualisation of test-set examples with high DSC scores.

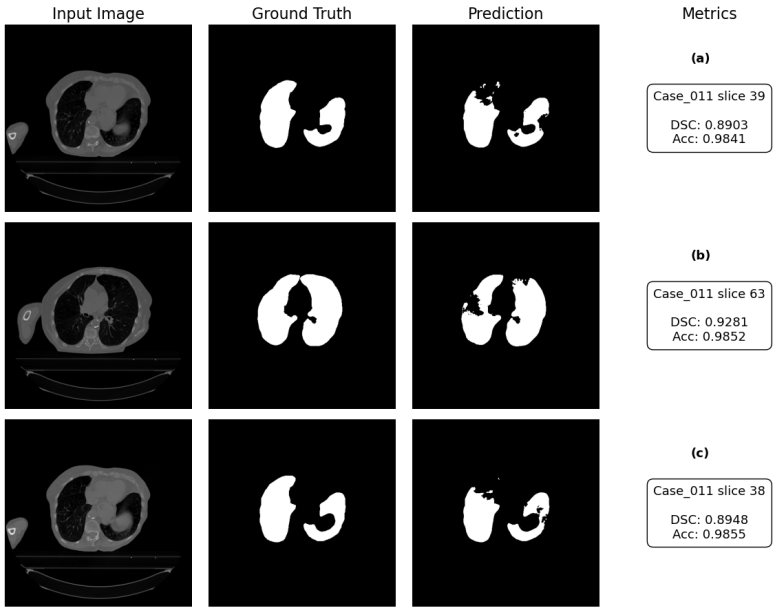

Figure 6: Visualisation of the lowest accuracy test-set examples (all from case 011).

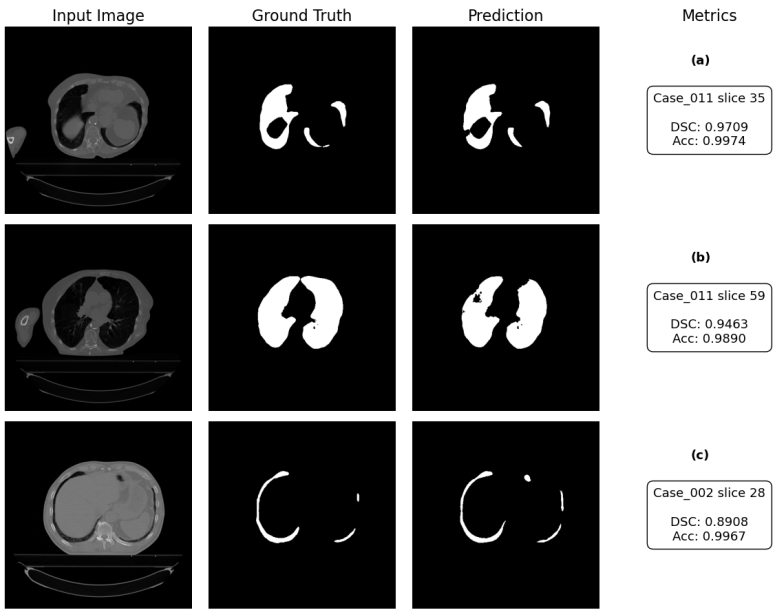

Figure 7: Visualisation of test-set examples with intermediate DSC scores.

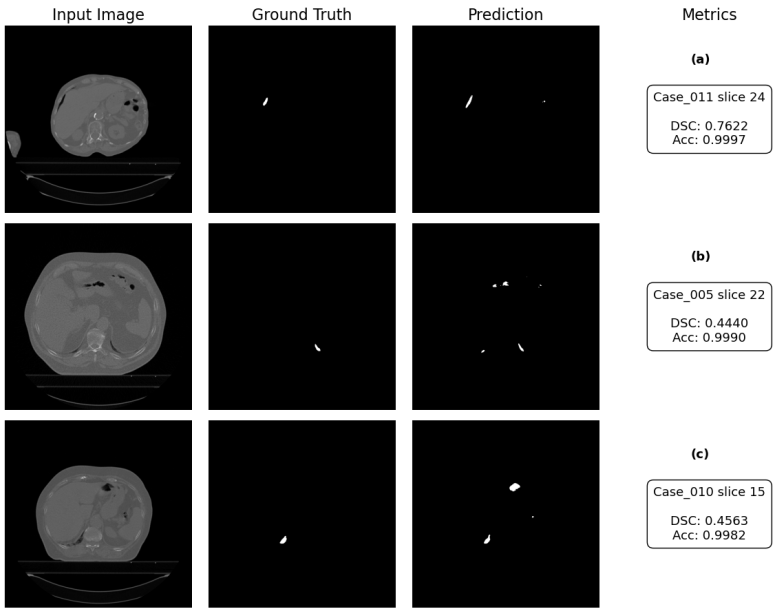

Figure 8: Visualisation of test-set examples with low DSC scores.

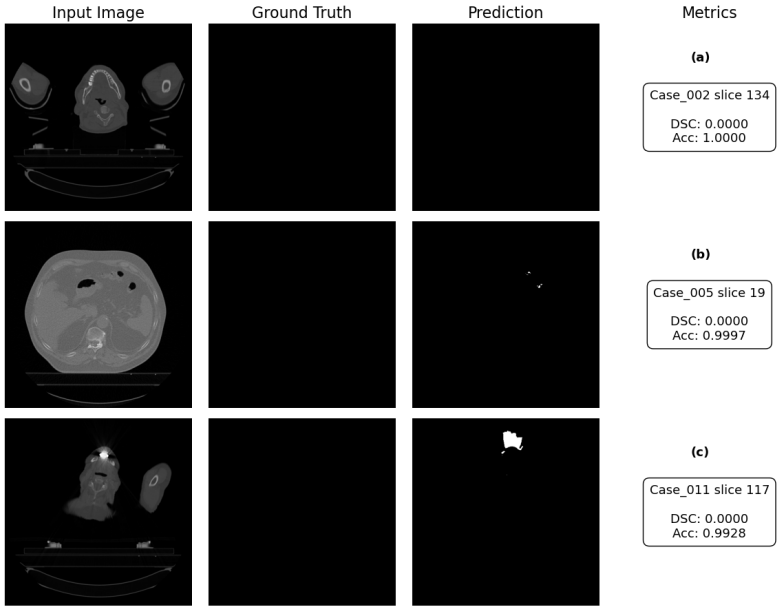

Figure 9: Visualisation of test-set examples with empty ground truth masks and (a) Perfect accuracy (b) Intermediate accuracy (c) Low accuracy.

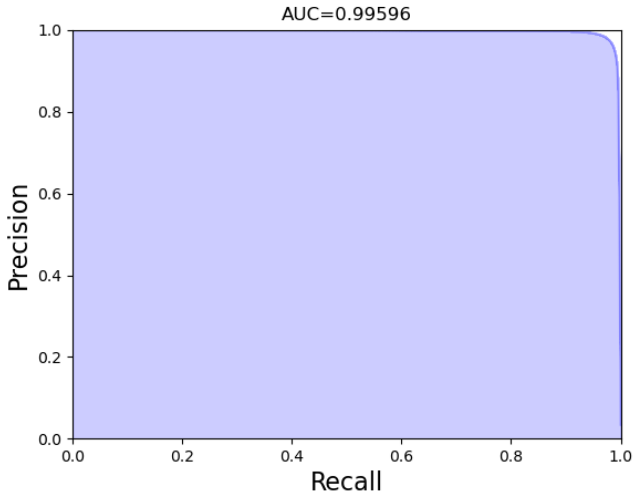

Figure 10: Precision-recall curve of trained model on test set, with an area-under-curve of 0.996.

(c) Discussion

Overall, the small U-Net model performed very well on the test set, with 78.2% of its predictions categorised as 'excellent', and an area under the PR-curve of 0.996. However, the results were far from perfect, with many of the images with small masks (top/bottom of the lungs) having mid-to-low DSC, and many images with empty-masks (above/below the lungs) having imperfect accuracy. This is likely due to the high variability in these regions, where there are many small dark spots that are difficult to differentiate from the lungs. Additionally, we found a set of low-accuracy predictions in Case 011 (Fig. 6), which show large gaps where the left lung is. One explanation is that the left lung of patient 011 is abnormally large, causing the model under-predicted its size.

Many of these issues could likely be remedied by improving the quality of the training data. Collecting more training data to reflect all the variability in the lung CT scans is likely the safest way to boost performance, but we could also perform data augmentation with transformations like flipping, scaling and saturation to make the model more robust to variations in the data. Furthermore, we could enhance the contrast of the images using windowing or histogram equalization [7], which may make it easier for the model to identify the lungs.

The model architecture and training strategy also have much room for improvement. For instance, we could train the model for many more epochs beyond 10 and employ early stopping, allowing the model to reach its full potential without over-fitting. This would involve tracking the model loss on a validation set (separate to the test set), and stopping training when the validation loss stops improving. We could also experiment with more sophisticated optimizers like AdamW, or implement a learning rate schedule with SGD, to improve and speed up convergence. Our calculation of specificity and recall in the test set revealed a possible bias of the model towards negative predictions. This could be improved by optimizing the α and β parameters in the loss function (equations (2) and (3)) to increase the contribution of the dice term and the penalization of false negatives. We used a small version of U-Net for computational efficiency, but it is likely that adding more layers or kernels to increase capacity of the model would allow it to learn more complex features, potentially leading to better results. Any of these hyper-parameters could be fine-tuned using efficient methods like Bayesian optimization [8], provided we have enough computational resources to perform such a search.

In summary, it was determined both quantitatively and qualitatively that the trained U-Net model produced consistent high-quality lung segmentations, although its performance could likely be refined further by improving the data, model architecture and training strategy.

References

[1] Ronneberger O, Fischer P, Brox T. (*2015*). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015; Cham: Springer International Publishing; p. 234-241.

[2] Yang J, Sharp G, Veeraraghavan H, et al. (*2017*). *Data from Lung CT Segmentation Challenge (LCTSC) [Data set]*. The Cancer Imaging Archive. doi:10.7937/K9/TCIA.2017.3R3FVZ08. Available at: `https://doi.org/10.7937/K9/TCIA.2017.3R3FVZ08`.

[3] DICOM Standards Committee. (*2024*). *Confidentiality Profiles in Table E.1-1 in PS3.15*. 2024a ed. Security and System Management Profiles. Available at: `https://dicom.nema.org/medical/dicom/current/output/html/part15.html#table_E.1-1`.

[4] Jadon S. (*2020*). *A survey of loss functions for semantic segmentation*. 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB); Via del Mar, Chile; p. 1-7. doi:10.1109/CIBCB48159.2020.9277638.

[5] Sudre CH, Li W, Vercauteren T, Ourselin S, Cardoso MJ. (*2017*). *Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations*. In: Deep learning in medical image analysis and multimodal learning for clinical decision support; p. 240–248. Springer. Available at: `https://pubmed.ncbi.nlm.nih.gov/34104926/`.

[6] Taghanaki SA, Zheng Y, Zhou SK, et al. (*2019*). *Combo loss: Handling input and output imbalance in multi-organ segmentation*. Computerized Medical Imaging and Graphics. 75:24-33. doi:https://doi.org/10.1016/j.compmedimag.2019.04.005. Available at: `https://www.sciencedirect.com/science/article/pii/S0895611118305688`.

[7] Lehr JL, Capek P. (*1985*). *Histogram equalization of CT images*. Radiology. 154(1):163-9. doi:10.1148/radiology.154.1.3964935. PMID:3964935.

[8] Snoek J, Larochelle H, Adams RP. (*2012*). *Practical Bayesian Optimization of Machine Learning Algorithms*. In: Advances in Neural Information Processing Systems 25; Editors: Pereira F, Burges CJ, Bottou L, Weinberger KQ. Curran Associates, Inc. Available at: `https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf`.

Appendix

A Auto-Generation Tools

Github copilot was used to assist with basic programming tasks like plotting figures. It was also occasionally used to assist with the generation of doc-strings of functions and files.