Group Technical Report

This study explores prostate cancer segmentation in magnetic resonance imaging (MRI). Using data from the PROMISE 12 Challenge, the study seeks to answer the following research question:

> *"Segmenting prostate glands from axial MR images can be performed in 2D or 3D volumetric data. Which one is better?"*

We created two convolutional neural networks for 2D and 3D segmentation, both based on the U-net architecture. The performance of each model was compared against each other using clinically significant metrics. This technical report provides links to access the code used throughout the project, outlines steps to reproduce results presented in the individual report, and provides a summary of the roles and contributions of each team member.

Code Submission

All the code is available in **Google Colaboratory** (Colab) as Jupyter notebooks. The trained models for 2D and 3D segmentation are stored in the following Google Drive folder. The U-Net structure was implemented using **Tensorflow 2.2.0**. These notebooks are also available in the zip folder submitted to Moodle. There are five Colab notebooks used in this study:

Function	Description	Colab Link	Filename (Zip folder)
Training Final 2D Model	Running this script will give the final optimized 2D model used in the comparison experiment. This model uses dropout and batch normalization, but without any data augmentation.	https://colab.research.google.com/drive/139qQbLI30ZU2eaVk1FH4Wr5wcjy6QPsA?usp=sharing	train_2D_final .ipynb
Training Final 3D Model	Running this script will give the final optimized 3D model used in the comparison experiment. This model uses dropout and batch normalization, but without any data augmentation.	https://colab.research.google.com/drive/1VHLmrbi92SOnCBlchYzGOAtqp307fL_E?usp=sharing	train_3D_final .ipynb
Training 2D Model with Data Augmentation	Running this script shows the results of training the 2D model with and without data augmentation.	https://colab.research.google.com/drive/1Xfz00M-7EO4zOXdNs89H1fiD1ZkKBmev?usp=sharing	train_2D_DA .ipynb
Training 3D Model with Data Augmentation	Running this script shows the results of training the 3D model with and without data augmentation.	https://colab.research.google.com/drive/1He2YxqqklG2iiINghWb6jrKVM-LcTvPx?usp=sharing	train_3D_DA .ipynb
Comparing 2D and 3D Models: Quantitative and Qualitative Analysis	Running this script provides the statistical test results included in the 'Individual Report' and also contains the overlay code for generating segmentation figures for qualitative analysis.	https://colab.research.google.com/drive/114HxZV-py8ei_RVhwEIoeEnseZ85ieBF#scrollTo=xrdK6NmEog7x	comparison .ipynb

Steps to Reproduce Results

This section outlines the ways to use the Colab notebooks to reproduce results in the 'Individual Report'. Steps to replicate the datasets used, model training, data augmentation and regularization, quantitative analysis, and qualitative analysis are discussed in the subsequent subsections.

Data

The PROMISE12 dataset has 50 MR images with segmentation ground-truths and a further 30 MR images without segmentation ground-truths. We used a pre-processed version of the dataset, provided by Dr. Yipeng Hu (see here). For simplicity, we used the 50 labeled images in this study. This dataset was randomly split into a train, validation, and test set that contains 40-5-5 images, respectively. There was no additional pre-processing performed on the data.

Model Training

The optimized hyperparameters used in both models are summarized in the table below:

Hyperparameters	Variable Name	2D Model	3D Model
Initial Learning Rate	learning_rate	7.5e-4	5.0e-4
Initial Number of Channels	num_channels[1]	32	16
Validation Size	val_size	1/9	1/9
Epochs	epochs	150	150
Batch Size	batch_size	128	4
Dropout	dropout	0.5	0.5

where **learning_rate** is the initial learning rate of the Adam Optimizer, **num_channels[1]** is part of an array **num_channels** that corresponds to the number of feature channels in the U-net architecture, **val_size** is 10% of the 50 MR images with ground truth segmentations (10% were excluded for the testing dataset, so 1/9 yielded the required 10% for the validation dataset), **epochs** are the maximum number of epochs used for training, **batch_size** was set to 10% of the training dataset (128/1280 slices for the 2D model and 4/40 images for the 3D model), and **dropout** was set to 0.5 (further discussed in the subsequent section).

To train the model, run the 2D training and 3D training Colab notebooks. After training has completed, the files to run TensorBoard are saved in the **logs** directory. The **saved_model** directory contains the model trained at the last epoch as well as the best performing model. We chose to use the best performing model for the comparison experiment.

Data Augmentation and Regularization

Regularization

Dropout was utilized after every layer in the U-net model, which was set to 0.5 in both 2D and 3D models to allow for an appropriate comparison. When increased to 0.6, model performance worsened slightly. More notably training was significantly slower, particularly for 2D. We used dropout as part of our baseline model, since we found that the networks struggled to train without it.

Batch normalization was applied before every activation layer in the model. This ensured that the activation layer inputs were normalized and centered to prevent oversaturation of the activation function. Parameters for batch normalization were:

Parameters	Value
Axis	-1
Momentum	0.99
Epsilon	0.001
Center	True
Scale	True

Data Augmentation

For the 2D model, 50% of image slices were copied and subject to rotations, horizontal and vertical flips, each with equal probability. Augmented images and their augmented corresponding labels were concatenated to back to the dataset

For the 3D model, 50% of the volumes were copied and subject to rotations along the z-axis and flips along all three axes. All augmentations had equal probability and the corresponding label was also augmented with the same parameters. The augmented volume was concatenated to the training dataset.

Run the 2D training with data augmentation and 3D training with data augmentation scripts to see that our data augmentation strategies did not significantly improve performance.

Quantitative Comparison Experiment

The following set-up was necessary before running the quantitative comparison experiment (see 'set-up' section in Compare_Unet_2D_3D):

- Load the pre-processed labeled dataset (from Dr. Yipeng Hu)
- Load the trained 2D and 3D models, which were trained with the optimized hyperparameters
- Make segmentation predictions on the testing dataset (5 MR images). These cases were the hold-out data, which was not used for training the models.

The comparison experiment was run using two different metrics - **Mean Absolute Distance (MAD)** and **Dice Similarity Coefficient (DSC)**. The score of the two models for both metrics is computed comparing the predicted masks from 2D and 3D models with the ground-truth labels respectively.

To calculate MAD scores, which is a boundary metric, we needed to exclude slices where either the label mask or predicted mask (for 2D and 3D separately) was empty. Out of 160 slices (32 slices each from 5 cases), **79** 2D slices and **77** 3D slices remained. MAD was calculated for each slice and the 2D and 3D scores were arranged in bins of ascending performance. These calculations were carried out in the first cell of section 'MAD t-test' in Compare_Unet_2D_3D.

DSC is a metric that evaluates the **overlap** between the labels and the predicted masks. DSC was calculated for each slice and the 2D and 3D scores were arranged in bins of ascending performance. The 'empty slices', which had DSC=1, for which both the labels and the masks only contained the background, were removed from the DSC t-test comparison. **99** 2D slices and **93** 3D slices were used. These calculations were carried out in the first cell of section 'DSC t-test' in Compare_Unet_2D_3D.

To compare the two models, the t-test was used to assess the difference between MAD score (2D versus 3D) and the difference between DSC score (2D versus 3D). We decided to split the results into 4 bins for the comparison experiment. Each quartile was compared using two-sample t-tests (alpha=0.05) and p-values were evaluated. The quartiles with the p-values lower than 0.05 showed a statistically significant difference in performances of either 2D or 3D models. The code for the statistical test is in the second cell of sections 'MAD t-test' and ''DSC t-test' in Compare_Unet_2D_3D.

Qualitative Comparison Experiment

The qualitative analysis code is also in the Compare_Unet_2D_3D script. It is designed to be run after the quantitative experiments and requires variables which are generated during the qualitative analysis. For this reason, it is recommended to follow all steps outlined in the **Quantitative Comparison Experiment** section of this report before attempting qualitative analysis.

The qualitative analysis focuses on identifying slides with large errors, particularly those with significant DSC results. There are two main categories of error to consider, firstly is the presence of false positives and false negatives, secondly is the presence of multi-blobs. DSC takes into account false positive and false negative results, so the qualitative analysis focused more on the latter cause of errors. Nevertheless, occurrences of false positives and false negatives (where either the model or the label slice are empty while the other contains pixels) are recorded and a graph displaying the location of these in each volume is produced in Compare_Unet_2D_3D. Slices which contain these errors are also noted in the output terminal as well as the total number of each error once analysis is complete.

To examine the slices which contain multi-blobs an overlay graphic for each slice is produced, with the outline of each mask overlaid onto the MR image they relate to. The label outlines are displayed in a thicker white line while the 2D and 3D model prediction mask outlines are drawn in green and red, respectively. These graphics allow slices with multi-blobs to be identified. These were identified manually, with three instances occurring in the label slices, 8 in the 2D model slices and 19 in the 3D model slices. There is only one occurrence of both the label and a model mask both featuring multi-blobs, which involves the 3D model mask, though only the larger blob overlaps in this case. A graphic displaying the MR image, the label and each model prediction individually was also created to display some examples of each error type and the code for this is included beneath the overlay graphics in Compare_Unet_2D_3D. 2D_Overlay_Masks and 3D_Overlay_Masks contain all the overlay figures for the 2D and 3D models, respectively. Instead of plotting the 2D and 3D model together (like in the Colab notebook), the figures in the hyperlinks separate the two models. These are the representative slices for the multi-blob error:

Model	MR Image Number (in test data)	Slice Number
2D	3	10
2D	5	20
3D	1	14
3D	2	14

Roles and Contributions

The roles and contributions of each team member are summarized in the table below:

Name	Contributions
Guglielmo Pellegrino	Set up of the Colab notebooks for the models training; design of the Colab notebook for the quantitative and qualitative analysis (comparison of the two models); dataset randomization and split; choice of the computational metrics.
Aman Ganglani	Architecture research and documentation along with relevance to medical research; model training using Tensorflow GPU; hyperparameter optimization and experimental design; visualization code for validation data and splitting validation data from training dataset.
Cyrus Tanade	Wrote original 2D and 3D segmentation code; performed 2D and 3D model hyperparameter search; trained final 2D and 3D models; designed quantitative comparison experiment; wrote 'Model Training' section of the group technical report and set the overall outline; maintained GitHub repository in the first half of the project before switching to Colab.
Jack Weeks	Data augmentation and regularization research and implementation; assisted in Colab setup; utilising tensorboard; assisted with bugs/architecture modifications.
Nikita Jesaibegjans	Computational metrics research; choice of metrics; designed qualitative comparison experiments of 2D vs 3D models; designed the Colab notebook to run the quantitative experiments on test data using the best models; 'Quantitative Comparison Experiment' section of the group technical report.
Josephine Windsor-Lewis	Project management; coordination of team members; research on the clinical relevance of quantitative and qualitative metrics; wrote qualitative analysis code and analysis of results; wrote mask overlay code for model comparison.