Machine Learning for Health Care

Project 1

Mesut Ceylan, Sven Gregorio, Moritz Vandenhirtz

March 2021

1 Introduction

Colon cancer together with rectal cancer ranks as the third most commonly diagnosed cancer with worldwide two million new cases in 2020 (Sung et al., 2021). Due to its high mortality rate it is very important to detect colon cancer as soon as possible. However, less than 50% of the cases are diagnosed at an early stage (Siminoff et al., 2015). As recent years have showed, Machine Learning can be extremely helpful in the healthcare sector because of its ability to process the increasingly vast amounts of data. Therefore, this project tries to utilize neural networks in order to improve the detection of colon cancer.

2 Methods

As input data, we are given 100 colon cancer CT scans with a varying number of slices in the depth dimension. To formulate a robust deep learning model, our group experimented with the U-Net deep learning architecture and its hyperparameters before deciding on the final model. Overall, we experimented with different loss functions, learning rates, dropout types and ratios, padding types and number of epochs to determine the best performing model. We depict some of the model performances in Table 1 and Table 2.

Mesut implemented the U-Net structure from Keras API. He formulated different loss functions and performance metrics such as Jaccard Distance Loss and Jaccard Index to meet the projects need. In the end the group decided to continue with the Binary Cross-Entropy Loss due to more stable training and convergence. He also down-sampled the image size using the bi-linear interpolation in order to get a faster training. While doing the exploratory data analysis, we observed that almost no details are lost if instead of 512×512 pixels, we only use 256×256. Also the ratio of positive pixels (roughly 5 percent) stays the same when images are down-sampled. We concluded that reducing the image size does not cause a loss of information, while it allows faster computations and prevents overfitting due to fewer necessary parameters.

Sven worked on solving the label imbalance, and tried the following three approaches. The first approach was to weight the labels by their inverse class frequency. The second approach was to split the class imbalance in two parts by using a two-stage model: the first stage would predict whether a slice had at least a positive label and the other would try to predict the positive labels with the assumption that at least one was positive. The second part of this model only needed to be trained on the small subset of the data with positive labels. We abandoned this approach because it was too complex. The third approach (used by our final model) is to use oversampling for inter-slice imbalance and class weights for intra-slice imbalance.

Moritz experimented with appending adjacent slices to the input slice in the standard U-Net. Depending on how many slices are appended, the model gets a better idea of the depth of the image. However, the amount of parameters also increases.

In order to choose our final model, each of us performed hyperparameter tuning using cross-validation by using 80 percent for training and 20 percent for validation of the whole dataset. A subset of the hyperparameters we experimented are depicted in the Table 2. As a group, we regularly meet over Zoom to discuss the implementations and projects.

3 Results

We analysed the validation performance of the models in terms of the following metrics:
Mean Intersection over Union(IoU) of positive class, Precision, Recall, Training Time and Binary Cross Entropy Loss.
We also compared the IoU to the baseline of uninformed guessing which would always predict 1.

	Learning Rate	Dropout Type	Dropout%	# of Epochs	Filter Type	Padding	Mean IoU
Model 4	1.00E-03	No dropout	0.0	40	Type1	Valid	0.080
Model 5*	1.00E-03	No dropout	0.0	40	Type1	Valid	0.096
Model 8	1.00E-03	Dropout	0.2	40	Type1	Valid	0.096
Model 7	1.00E-03	No dropout	0.0	100	Type1	Valid	**0.121**

Table 1: Top performing models. Note that Dropout refers to Dropout layers within the first layers of the U-Net architecture. Also note that Type-1 Filter refers to filter size of [64, 128, 256, 512] in contraction and [256, 128, 64, 32] in expansion sections. Finally, * refers to maintaining anti-aliasing parameter of resizing function set to True. The best performing model however, neither utilizes Dropout layers nor the anti-aliasing. The best model is trained for 100 epochs which eventually made the biggest performance difference. For extended hyperparameter search, please refer to Table 2.

After evaluating the performance of the different models, we decided to use the following architecture for our final prediction: We cut the images into slices and downscale each to a resolution of 256×256 pixels with bi-linear interpolation. To tackle the problem of the highly imbalanced data set, we utilized oversampling for positive labels. This means that we categorize the slices of each scan into "contains at least one positive label" and "contains no positive labels". Then, we trained the model using the same amount of slices from both classes in each batch. This approach allowed us to solve the problem of inter-slice class imbalance. To tackle the intra-slice imbalance, we used class weights which are taking into account that 50% of the inputs that the model gets are slices without any tumour. After this preprocessing method, we use an adapted version of the U-Net which takes the lower resolution into account. Having solved the problem of extremely high class imbalance, we can now use the **Binary Cross Entropy Loss** and do not need to resort to problem specific loss functions. We benefited from early stopping to save model checkpoints and trained the network for **100 epochs, Adam optimizer with learning rate of 1e-3 and batch size of 4**. After training, we achieved the **mean Intersection over Union on the positive labels of 0.121** on the validation set. During training, we utilized Tensorboard callbacks to monitor the model performances.

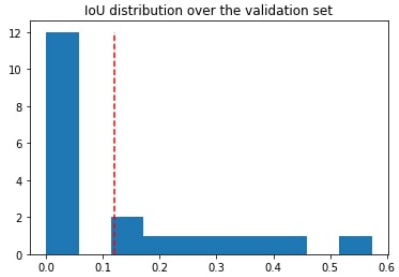

Figure 1: Validation IoU

In figure 1 we show the histogram of the positive IoU of every image scan in the validation set. We observe that there is high variance between image scans. One explanation might be that our model only learnt specific invariances and body shapes and struggles to generalize. Another might be that by chance we picked a hard validation set. We observe that for half the images we perform really well. Future research might look into how we can improve the other half.

References

Siminoff, L. A., Rogers, H. L., and Harris-Haywood, S. (2015). Missed opportunities for the diagnosis of colorectal cancer. *BioMed research international*, 2015.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F. (2021). Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*.

Appendix

	Learning Rate	Dropout Type	Dropout%	# of Epochs	Filter Type	Padding	Mean + IoU
Model 1	1.00E-03	Spatial2D Dropout	0.5	40	Type1	Valid	0.019
Model 2	1.00E-03	Spatial2D Dropout	0.5	40	Type1	Valid	0.045
Model 3	1.00E-03	Spatial2D Dropout	0.2	40	Type1	Valid	0.045
Model 4	1.00E-03	No dropout	0.0	40	Type1	Valid	0.080
Model 5*	1.00E-03	No dropout	0.0	40	Type1	Valid	0.096
Model 6	1.00E-03	No dropout	0.0	80	Type1	Same	0.069
Model 7	1.00E-03	No dropout	0.0	100	Type1	Valid	**0.121**
Model 8	1.00E-03	Dropout	0.2	40	Type1	Valid	0.096
Model 9	1.00E-03	Dropout	0.2	80	Type1	Valid	0.030
Model 10	1.00E-03	Dropout	0.5	40	Type1	Valid	0.004
Model 11	2.00E-04	Dropout	0.2	40	Type1	Valid	0.058
Model 12	2.00E-04	Dropout	0.2	80	Type1	Valid	0.062
Model 13	2.00E-04	Dropout	0.2	40	Type2	Valid	0.069
Model 14	2.00E-04	Dropout	0.2	80	Type2	Valid	0.054
Model 15	2.00E-04	Dropout	0.2	40	Type2	Same	0.069
Model 16	1.00E-03	Dropout	0.2	80	Type2	Same	0.062

Table 2: The extended hyperparameter search study. Please note that Dropout layers are utilized in the first layers of the model architecture while Spatial Dropout2D layers are used within first layers of the U-Net as well as within Contraction and Expansion sections. Also note that Type-1 filter refers to filter size of [64, 128, 256, 512] in contraction and [256, 128, 64, 32] in expansion sections while Type-2 refers to down-scaled filter sizes by 4 times. Finally * refers to maintaining anti-aliasing parameter of resizing function set to true. The best performing model however, neither utilizes Spatial Dropout2D layers nor the anti-aliasing. The best model is trained for 100 epochs which eventually made the biggest performance difference.