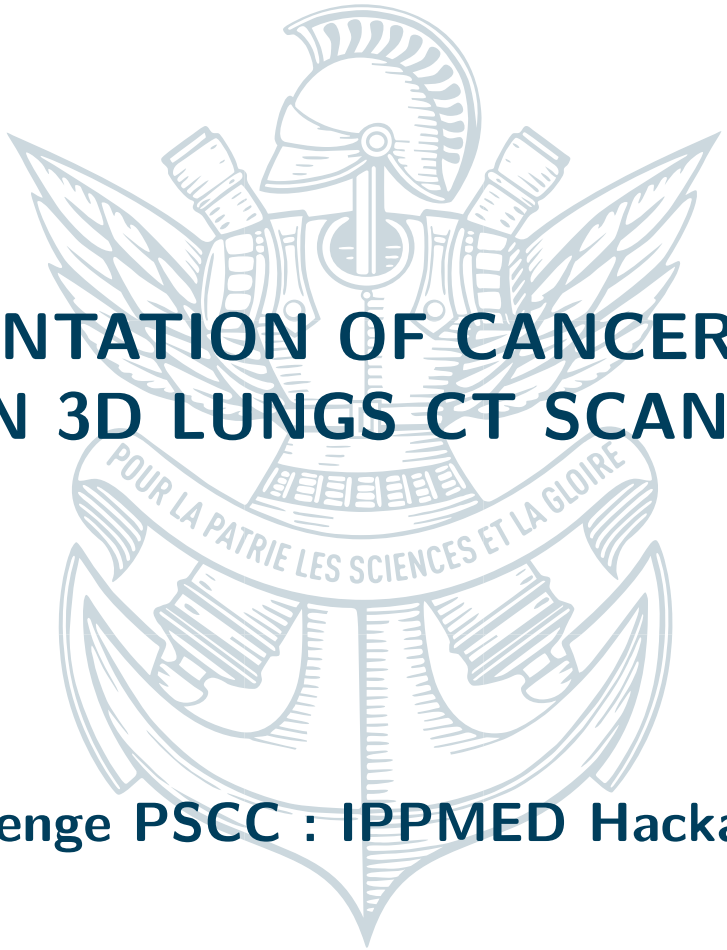

SEGMENTATION OF CANCER CELLS IN 3D LUNGS CT SCANS

Challenge PSCC : IPPMED Hackathon

30 January 2024

Thomas Loux

ÉCOLE POLYTECHNIQUE

IP PARIS

1 LUNGS CANCER AND CT SCANS

1.1 CT scans and Nifti files

The data is provided in two folders, one for the training data and one for the test data, with 291 training images and 100 test images. With the training dataset, we also have masks for cancer cells. The masks are also in Nifti format, but they are binary images, with 0 for the background and 1 for the cancer cells. The goal of the challenge is to segment the cancer cells in the test images, and to submit the results in the same format as the masks.

A first analysis of the scans shows that the shape is 512x512xN where N is around N. Nonetheless, the voxel size is always the same (around 1mm x 1mm x 3mm). Hence the difference relies on the number of slices to obtain the complete scan of the upper body. The images are anisotropic but it is not really a problem if the model is consistently train with the same voxel size. The images are also grayscale, with a value between -1024 and 3071. The value -1024 corresponds to the background, and the value 3071 corresponds to the bones. The values come from absorption of X-rays by the body. The more the X-rays are absorbed, the brigther the pixel is.

1.2 Masks and loss

All the image is not relevant for the segmentation, we can see the bowl and the neck. Moreover there is also some void around the body, hence it is possible to crop the image to reduce the size of the image.

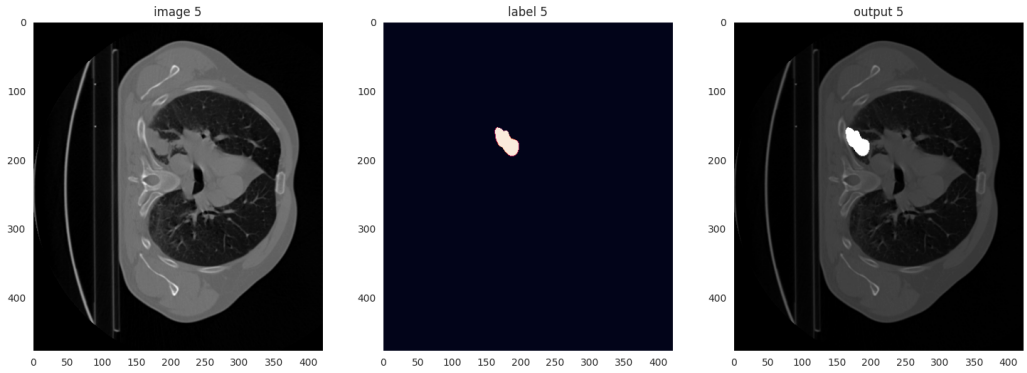

Figure 1: Example of a scan and mask

So we have a mask for the training, hence the training is supervised. There is also a rule in the challenge that forbids to use a pretrained model. Hence we cannot use a pretrained model for the segmentation, that otherwise would have been a very good start as the use case is not unsual.

Regarding the loss, we can use the binary cross entropy loss, as the mask is binary. We can also use the dice loss, which is a loss that is often used for segmentation. It is defined as follows :

$$\text{Dice Loss} = 1 - \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2} \tag{1}$$

where p_i is the prediction and g_i is the ground truth. The dice loss is a good loss for segmentation as it is not sensitive to the class imbalance. Indeed, the number of pixels that are cancer cells is very small compared to the number of pixels that are not cancer cells. Hence, the loss would be dominated by the pixels that are not cancer cells. The dice loss is not sensitive to this problem.

The dice can also be rewritten as follows :

$$\text{Dice Loss} = 1 - \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{2}$$

$$\text{Dice Loss} = 1 - \frac{2|A \cap B|}{|A| + |B|} \tag{3}$$

$$\text{Dice Metric} = \frac{2|A \cap B|}{|A| + |B|} \tag{4}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives. The Dice loss is effective for segmentation because it is less sensitive to the class imbalance problem. Indeed, the number of pixels that are cancer cells is very small compared to the number of pixels that are not cancer cells. Hence, the loss would be dominated by the pixels that are not cancer cells for only Cross Entropy loss. However, some papers show that the convergence is faster and achieve better results with the sum of the two losses. This loss is called DiceCE.

Moreover, the challenge also use the relative error on the volume predicted and also the RECIST error. The RECIST error is a measure of the error on the diameter of the tumor. It is defined as follows :

$$\text{RECIST relative error} = max(\frac{|d_{\text{pred}} - d_{\text{gt}}|}{d_{\text{gt}}}, 1) \tag{5}$$

where d_{pred} is the diameter predicted and d_{gt} is the ground truth diameter. We choose the largest diameter of the volume.

The gradient of these errors are not the same as the gradient of the dice loss so the penalization is not the same. Particularly, it heavily penalizes the overestimation of cancer cells.

For the rest of the project, the problem will be considered as a multiclass segmentation problem. This approach is more general and can be applied to other problems. Moreover, it enables us to avoid the use of a thresold. The problem is then to predict the probability of each class for each pixel, here either the background or the cancer cells.

2 UNET AND SEGMENTATION

2.1 UNet architecture and inference strategy

The UNet architecture is a convolutional neural network that is often used for segmentation. It is composed of a contracting path and an expansive path. The contracting path is composed of convolutional layers with a kernel size of 3x3 and a stride of 1. The number of channels is doubled at each layer. The purpose of this network is create masks or images from an input. The contracting path allows to extract features a differents scales. The network can also be extended to 3D.

The prediction can be made in two ways. The first one is to predict the whole image at once. The second one is to predict the image by splitting it into patches. The second one is more memory efficient, but it is also slower. The first one is more memory consuming, but it is faster. The second one is also more accurate as the network can see more context.

After the test of basic Unet, we also tried some more advanced architectures with attention mechanisms. We decided to use SwinUnetr based on papers and on the most classic nets used in the field. The use of this model boosted our performance although the validation dice loss was about the same as the Unet. We suspect that the model miss prediction were better, at least suggesting some more realistic tumors in terms of size and aspects. Yet we did not investigate thoroughly this.

The computations are done using SLURM and with Distributed Data Parallel from PyTorch which allows multi GPU training, mono or multi node. The training with multi node was unstable so we choose to only use 2 A40. The training was really faster with the 2 A40 (which was not the case with the Data Parallel, another distributed scheme from PyTorch). It is simpler but not as flexible.

The code was implemented using PyTorch and the framework Monai [1]. Monai provides builtin functions, preprocessing and post processing for medical images, as well as networks and losses. Knowing the complexity of vision networks and transformation, this framework is really useful.

For the inference and the patch strategy, Monai provides a sliding window inferer. The 3D images is cropped into overlapped patches, and the network is applied to each patch. The patches are then reconstructed into a single image. The overlapped patches are then averaged. We show that the performance was better with higher overlap. We choose to use a high value of overlap of 0.8 that demonstrated of gain of about 0.03 of loss compared to an overlap of 0.5. Yet the inference time is subsequently increased.

For the final mask, Monai provides a easy way to inverse any preprocessing transformation that was applied to the image. Hence, we can easily inverse the cropping, and finally take the softmax to obtain a binary mask.

For the preprocessing, we tried to crop some part of the image using some thresholding on the image. Finally we deciced not to use it because it did not decrease the inference time. We use some augmentation techniques that were meant to follow the symetries of our problem : rand flip left-right, rand shift intensity and rand scale intensity.

For the post-processing, we tried numerous approaches. The first one was to take the biggest connected component. It was by far the most efficient. Furthermore, the approach was coherent with the true masks. But we manage to improve the score by focusing on the border of the mask. We did a postprocessing with the output from the model. We looked at the largest component predicted by the model. For this component C, we computed binary erosion to get a reduced comoponent C-eroded and binary dilation to get an augmented component C-augmented. Then we rejected voxels in C-C-eroded that unusual intensity and accepted voxels from C-dilated-C that had the right intensity. To see if a voxel had the right intensity, we computed the mean and variance of the largest component and made the modification based on a confidence interval, excluding voxels on the border of the largest component far from the mean and including voxels outside of the component but close from the mean

2.2 Training

For the patch approach, we use a convenient transformation from Monai that allows to crop in training image patches while respecting some proportion for the number of patches with positive or negative labels. The proportion is set to 0.25, which means that a quarter of the patches will have positive labels and three quarters of the patches will have negative labels. The patches are of size 160x160x64.

The training was done with a batch size of 1, a learning rate of 1e-3 and a diceCE loss. The training was done for 300 epochs. The training was done on the A40 for around 20 hours.

3 RESULTS AND DISCUSSION

3.1 Results

The patch strategy proved to be quite effective. The validation metric is quite unstable but it seems to be common. We reach at the end of the training around 0.5 of dice metric on the validation set.

The results are also quite good, the model is able to learn and to predict the cancer cells. The plot suggests that the training was too short, as the training has not plateaued yet. With an analysis of the predicted mask, we can see that there is artefacts from the sliding window. The artefacts are more visible because some parts of the mask are squarred. It is possible to reduce the artefacts by increasing the overlap between the patches and apply a gaussian filter on the patches to decrease the contribution of the edges.

On analysis of the evaluation results shows that the variability of the model is great. It can achieve 0.7 on the dice metric but also 0, when it predicts cancer cells on the wrong part of the scans.

4 CONCLUSION

This very challenging hackathon is a great way to discover the implementation of deep learning models for medical images. We took a real pleasure to work on this project.

REFERENCES

[1] M. Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Andriy Myronenko, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng

Tang, Isaac Yang, Michael Zephyr, Behrooz Hashemian, Sachidanand Alle, Mohammad Zalbagi Darestani, Charlie Budd, Marc Modat, Tom Vercauteren, Guotai Wang, Yiwen Li, Yipeng Hu, Yunguan Fu, Benjamin Gorman, Hans Johnson, Brad Genereaux, Barbaros S. Erdal, Vikash Gupta, Andres Diaz-Pinto, Andre Dourson, Lena Maier-Hein, Paul F. Jaeger, Michael Baumgartner, Jayashree Kalpathy-Cramer, Mona Flores, Justin Kirby, Lee A. D. Cooper, Holger R. Roth, Daguang Xu, David Bericat, Ralf Floca, S. Kevin Zhou, Haris Shuaib, Keyvan Farahani, Klaus H. Maier-Hein, Stephen Aylward, Prerna Dogra, Sebastien Ourselin, and Andrew Feng. Monai: An open-source framework for deep learning in healthcare, 2022.

[2] Matthias Hadlich, Zdravko Marinov, and Rainer Stiefelhagen. Autopet challenge 2023: Sliding window-based optimization of u-net, 2023.

[3] Andriy Myronenko, Dong Yang, Yufan He, and Daguang Xu. Automated segmentation of organs and tumors from partially labeled 3d CT in MICCAI FLARE 2023 challenge. In *Submitted to MICCAI 2023 FLARE Challenge*, 2023. under review.