ML4HC Project 1 Report

Juan Bermeo, Elia Saquand, Adrian Seiterle, Malgorzata Speichert

Overview

In this project, we trained a U-net to identify colon cancer in CT scans. As training data, we have one hundred CT scans, comprising multiple cuts. For every cut we received a mask image as shown in Fig. 1.

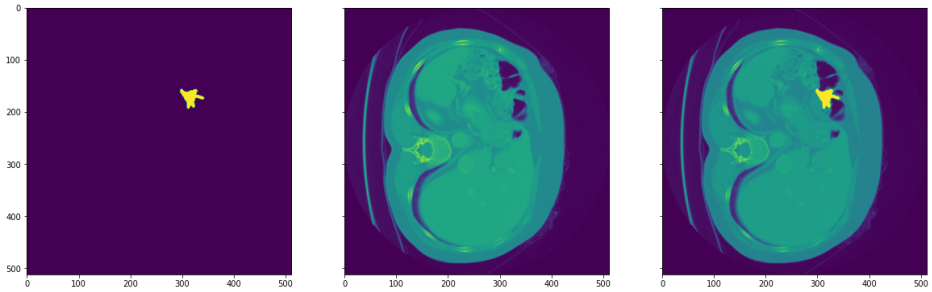

Fig. 1: Mask with the cancer section masked (left), the corresponding cut from the CT image (middle), mask overlaid on the cut from the CT scan (right).

Implementation

We used an Unet2D implemented with Keras. To get 2D images, we extended the `tf.keras.utils.Sequence` class to load the 3D scan and slice it along the z-dimension. Using the same class, we implemented a series of augmentation transformations with OpenCV and a logic to up and down sample images with and without cancer, respectively, given the severe class imbalance in this task. To build our model, we have used the function custom_unet from the keras_unet helper package. As the size of the dataset is relatively small in proportion to the number of parameters of a common Unet implementation like that of pix2pix, we used a smaller Unet with 32 filters in the first level - instead of 64 - and four levels of depth instead of 7. We used a dropout layer between every convolutional layer pair of the encoder with a rate of 0.2 to further avoid over-fitting (see custom_unet diagram). The activation layer at the output is a sigmoid function for each pixel. Finally, we used a binary focal loss to both give more weight to the positive class and to give less weight to correctly classified pixels (as they are mostly trivially correct zeros)

Preprocessing:

To iterate faster during hyper-parameter tuning, we resized our images to (128, 128) instead of (512, 512). We clipped the HU values of the pixels to min = -1000 and hounsfield_max = 400, as we found reports that only bone tissue has HU units greater than 400. For data augmentation, we implemented on-the-fly transformations - for each image, one of the augmentation or identity transformations was chosen at random - like rotation (± 30°), vertical and horizontal flipping, and random cropping, as is done in [1, 2].

We noticed that 95% of images have at least one cancer pixel in at most 21% of slices. In relation to the volume, in 99% of images, the cancer pixels make up less than 0.3% of the total volume. Because of this, we implemented a more aggressive down-sampling and up-sampling of 0.4 and 2 respectively.

Evaluation:

To calculate the IoU, we predicted on 2D mini batches of 128x128, resized to 512x512, and then restacked them to get a 3D image to compare it against the label mask in its original size to get the IoU on images of the holdout set.

Hyperparameter tuning:

We chose a 90/10 split and planned on using 3 out of the 10 folds to have more accurate estimates. We tuned the following hyperparameters: loss function (cross entropy loss, dice loss, jacquard loss, and focal loss), weights of class sampling, learning rate, types of augmentation transformations, and the number of filters at the first level. Out of these, what had the highest impact was the choice of loss function (binary focal loss) and up and down sampling. All experiments that did not use these two resulted in a negligible IoU in the holdout set, even after tuning the classification threshold, after training for 20 epochs. A higher number of epochs were rarely tested because of the long computation time, even on GPUs.

Our eventual choice of hyperparameters was:

- Binary Focal Loss with a gamma of 2 as recommended in [3] and alpha of 0.7. More aggressive gammas and alphas slowed considerably the convergence.
- Adam optimizer with a scheduled learning rate.
- Augmentation: rotation (± 30°), vertical and horizontal flipping, and random cropping (0.8, 0.9).
- Up sampling of 2 and down sampling of 0.4 for the positive and negative class.
- Capping of HU at hounsfield_min = -1000 and hounsfield_max = 400. Resizing each slice to (128x128).
- Trained for 17 epochs.

Results

The IoU of this choice on a single holdout set was of 0.0176 after using a threshold of 0.0875.

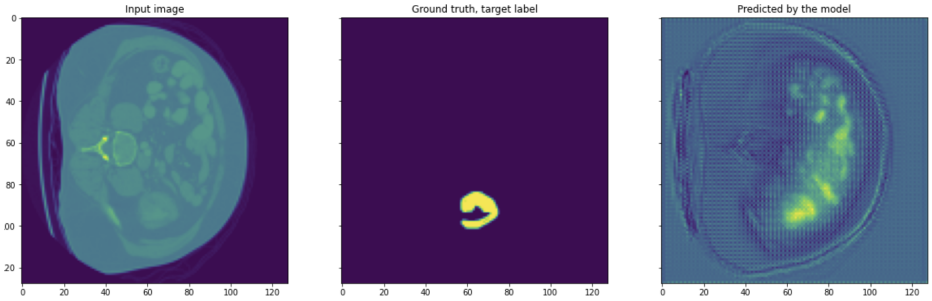

Fig. 2: Prediction on image slices that had cancer

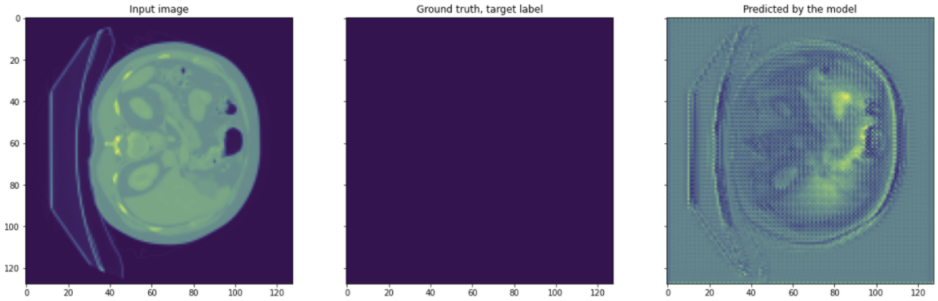

Fig. 3: Prediction on image slices that did not have cancer

Alternatives: We have also tried to implement a 3D Unet with pyTorch using DiceLoss which gave us an IoU of 0.0011.

Individual contributions: We pretty much all worked together on the whole implementation (Preprocessing, building Unet, Hyperparameter tuning) plus the writing of the report.

References

[1] Lin, T., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018, February 07). Focal loss for dense object detection. Retrieved March 22, 2021, from https://arxiv.org/abs/1708.02002

[2] Sandfort, V., Yan, K., Pickhardt, P., & Summers, R. (2019, November 15). Data augmentation using generative adversarial NETWORKS (CycleGAN) to IMPROVE generalizability in CT segmentation tasks. Retrieved March 22, 2021, from https://www.nature.com/articles/s41598-019-52737-x

[3] Zeshan, H, et. al.. (2018, April). Differential Data Augmentation Techniques for Medical Imaging Classification Tasks. Retrieved from https://europepmc.org/backend/ptpmcrender.fcgi?accid=PMC5977656&blobtype=pdf