“BIOMEDICAL COMPUTER VISION COURSE”

Segmentation challenge (CHOS19)

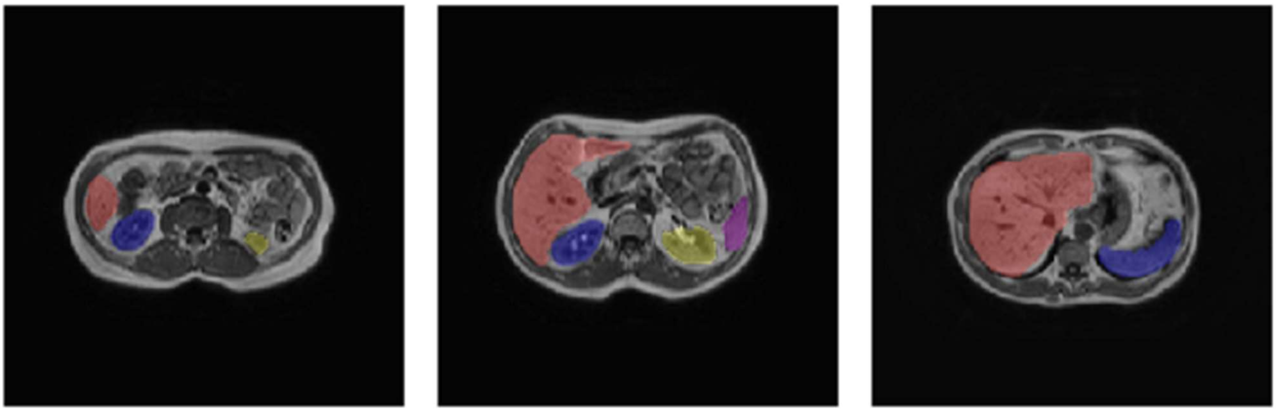
Gabriele Lorenzetti

Politecnico di Milano

Introduction

CHAOS challenge aims the segmentation of abdominal organs (liver, kidneys and spleen) from CT and MRI data and i chose to use only MRI (T1DUAL) due to the limited space on Google Drive.

An exemple

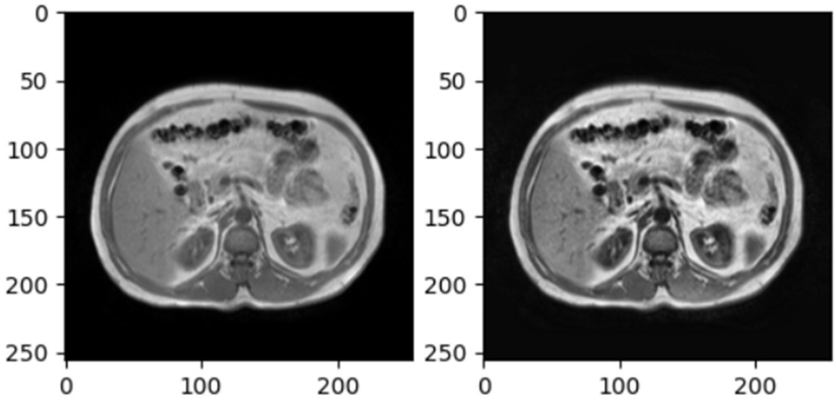


Preprocessing

In this first phase I set my self the goal of standardizing all the images and mask to a (256,256,256) shape doing a resampling first and a resize afterwards, the images have also been normalized and equalized for better contrast.

During the preprocessing I created a custom dataset on Google Drive (splitting it in train and validation, 80:20) to free the limited RAM of Google Colab and to work better with neural network.

An exemple of equalization



Neural Network

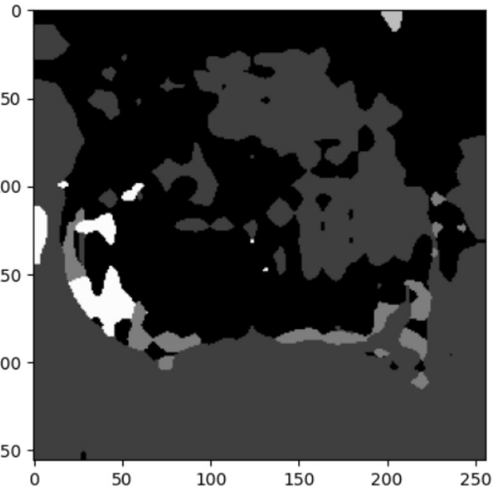
I implemented the network through the nn module of Pytorch and built the model using Resnet101 network (originally created to solve image classification task), used as a pre-training and bringing it to network for segmentation by convolution.

To calculate the loss, I decided to use the Cross Entropy function and to keep track of the results in a log.csv file with Adam as the optimizer, while the accuracy and precision metrics with f1score.

Results

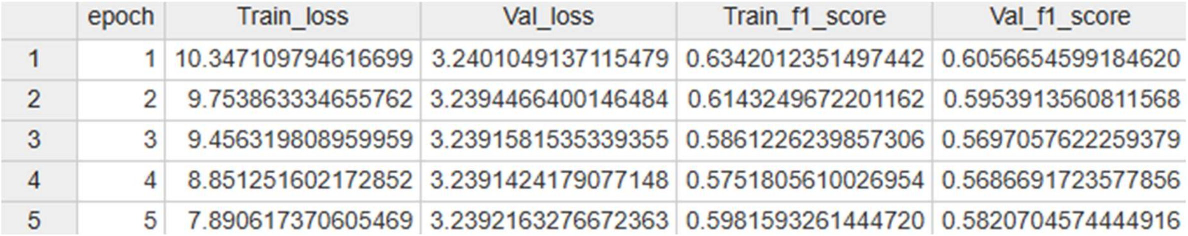
The results obtained are not even remotely close to those expected. This certainly depends on the few eras used, only five.

Another cause could be the chosen network architecture, maybe the Resnet is not good for segmentation; a solution could be the implementation of a u-net or maintain the first levels of the Resnet.



At the side is a result, viewing several of them I noticed that the network is starting to slightly understand the outline of the abdomen.

Below there is the log.csv file. We can see that the loss tends to decrease, which is a positive sign even if it is high



	epoch	Train_loss	Val_loss	Train_f1_score	Val_f1_score
1	1	10.347109794616699	3.2401049137115479	0.6342012351497442	0.6056654599184620
2	2	9.753863334655762	3.2394466400146484	0.6143249672201162	0.5953913560811568
3	3	9.456319808959959	3.2391581535339355	0.5861226239857306	0.5697057622259379
4	4	8.851251602172852	3.2391424179077148	0.5751805610026954	0.5686691723577856
5	5	7.890617370605469	3.2392163276672363	0.5981593261444720	0.5820704574444916