CEREBRAL ARTERIES SEGMENTATION TO GUIDE ENDOVASCULAR TREATMENT OF ISCHEMIC STROKE

Pilotto Loris

École Polytechnique Fédérale de Lausanne - Data Science semester project

ABSTRACT

This paper describes the CNN methods used in the case of brain's arteries segmentation. This project has been offered by the MIP:lab, led by professor Van De Ville Dimitri. It has been supervised by Dr. Maria Giulia Preti & Dr. Hofmeister Jérémy.

Index Terms— Arteries segmentation, U-Net

1. INTRODUCTION

New technologies allow doctors to perform increasingly complex tasks with the goal of helping patients, but also to minimize the effects of any medical intervention. In the case of strokes, the medical team has no time to lose and has to intervene with a surgical precision. With the development of fast and efficient deep learning models, their usefulness in the case of stroke seems motivated. The goal of the project was then to evaluate the effectiveness of a deep learning model in order to segment the brain's arteries based on 3D CT scans.

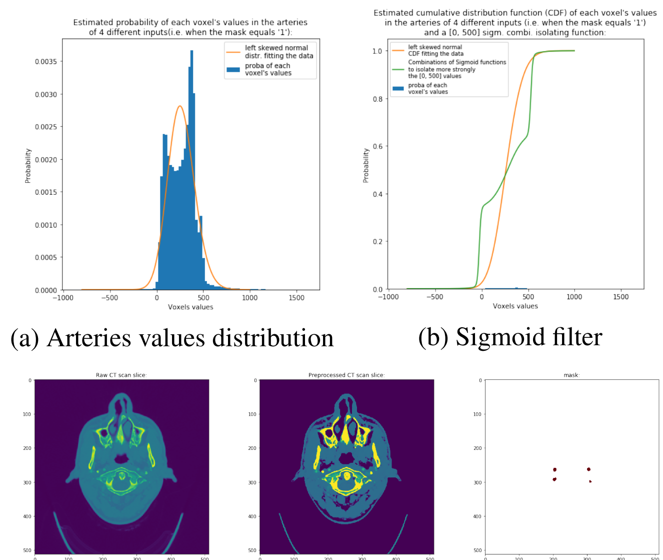
2. METHODS

2.1. Available data

For this project, five 3D greyscale CT scans of size Zx512x512 (where $Z \in [1000, 2500]$) and their corresponding brain arteries masks were available. With less than 0.02% of arteries among the voxels, this dataset is, as expected, highly unbalanced. Before taking the scans, a contrast agent (Accupaque 350) has been injected in the arteries of the patient. It facilitate the visualization of the anatomical structure of arteries.

2.2. Data preprocessing

The voxels values lied between -1000 and 1500 but the tremendous majority of arteries voxels' values were between 0 and 500 (see fig. 1 (a)). To distinguish this range of values, a sigmoid approximation of the arteries values' cumulative distribution function (CDF) was used with, in addition, two sharp sigmoid thresholds at value 0 and 500 (see fig 1 (b)). You can refer to fig. 1 (c) to see the effect of applying this sigmoid filter. Also note that even if this filter strongly highlighted the interval of values corresponding to arteries voxels, it is a reversible function, meaning that no information is lost by applying this filter.



(a) Arteries values distribution (b) Sigmoid filter

(c) Results on a CT scan slice

Fig. 1. Data preprocessing to highlight arteries.

2.3. Neural Network architecture

A lot of machine learning models have been designed for the specific task of image segmentation. The majority of these models take advantage of convolutional layers. Without doubt, the most famous CNN is the U-Net[2]. Its applications are numerous and its principles are easy to understand. In addition, it can perform accurate segmentation without having to rely on a huge amount of data. A U-Net architecture consists of two parts: a first contracting part where the input image goes through several iterations of convolutional layers followed by a reduction of its size (using a max pooling layer). The second part of the model iteratively restore to the image its original dimension (through upsampling or trans-

Thanks to MIP:lab for accepting my project.

posed convolutional layers). Before each downsampling, the last layer is directly connected to its upsampling corresponding layer (see fig. 2) to keep as much information from each image size as possible.

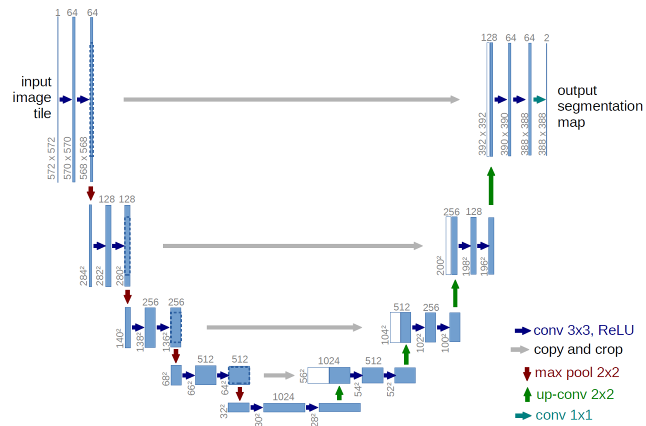


Fig. 2. U-Net architecture[3]

In this paper, the author focused on the implementation and optimization of different variants of U-Nets. As a starting point, a simple 2D U-Net with 520’549 trainable parameters (and an image size reduced to 256x256) and a more complex U-Net, with 2’622’677 parameters (and a image size reduced down to 32x32), were trained and tuned. In a second time, motivated by the good results of the two architectures, a third model has been tested where the simple and the complex U-Nets were trained in parallel (fig. 3). Finally, the same models

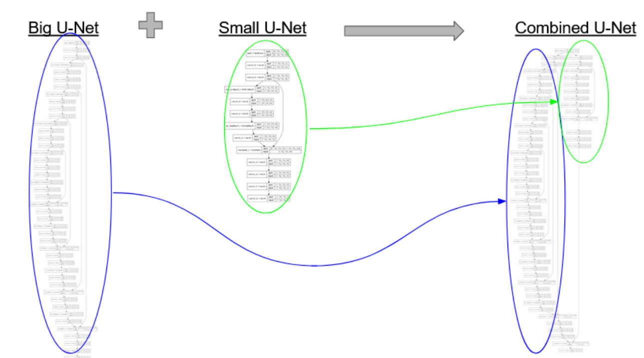


Fig. 3. U-Nets in parallel

were tested with 3D inputs (increasing the number of trainable parameters up to $\sim$ 7 millions).

2.4. Experiment

With 5 CT scans available, the choice of 4 images for training and 1 for validation seems the only viable solution. The function used to train the models was the dice coefficient (equivalent to the F1 score[1]):

$$Dice = \frac{2 * Intersection}{Intersection + union}$$

The function used as a metric was the Intersection over Union (IoU) that we want to have as close to 1 as possible:

$$IoU = \frac{Intersection}{Union}$$

3. RESULTS & DISCUSSION

$Remarque$: To be as objective as possible, only the metric ($i.e$ the IoU) value of the validation image was checked when comparing the models.

During this project, a basic model was first trained and then tuned to improve its performances.

3.1. 2D U-Nets results

A first basic U-Net was trained for 50 epochs. After 25 epochs, it’s validation IoU mean stabilized to 0.32 (see fig. 4) while the training IoU reached 0.7 and was still growing. A training metric more than twice as big as the validation one means that the model is overfitting. To fight this overfitting, drop layers were added to the model. The validation IoU increased to 0.36. Another method used to reduce this overfitting is data augmentation (random flipping, rotation and cropping of the input images). The validation IoU went from 0.32 to 0.39 once the drop layer and data augmentation were applied. Finally, the architecture of the U-Net was changed for the combined U-Nets architecture (fig. 3). This gave the best validation IoU of 0.41.

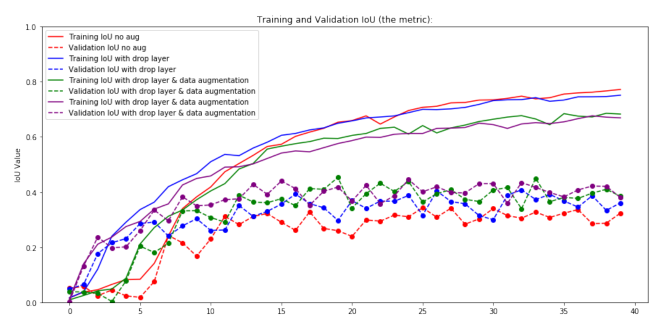


Fig. 4. 2D U-Nets IoU metric evolution

3.2. 3D U-Nets results

The 3D models were implemented as close as possible to the 2D U-Nets. The only change was the fact that now, the models take 3D images as inputs. In this paper, inputs with a third dimension equals to 2 or 4 are tested. A first basic model with third dimension of 2 gave a IoU validation metric of 0.35 (see fig. 5). Then, two models with drop layers, data augmentation and combined U-Nets architectures (but one with a 3^{rd} dimension of size 2 and one of size 4) were tested. With the Z dimension of size 2, the validation IoU reached 0.38 while the model with Z=4 reached 0.36.

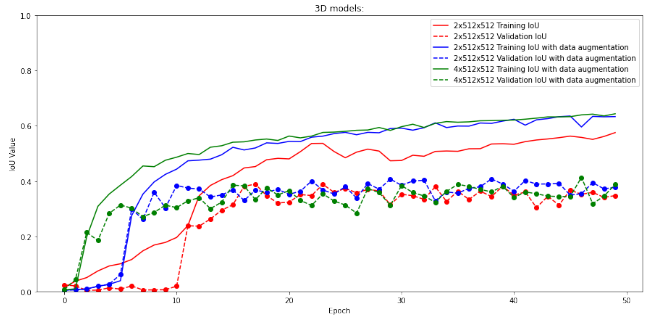


Fig. 5. 3D U-Nets IoU metric evolution

3.3. Results comparison

	2D Validation IoU
Basic U-Net	0.32
U-Net with drop layers	0.36
U-Net with drop layers & data augmentation	0.39
Combined U-Nets with drop layer & data augmentation	0.41

(a) 2D models validation IoU

	3D Validation IoU
Basic U-Net	0.35
Combined U-Nets with drop layers, data aug. & Z=2	0.38
Combined U-Nets with drop layers, data aug. & Z=4	0.36

(b) 3D models validation IoU

Fig. 6. 2D & 3D validation IoU results.

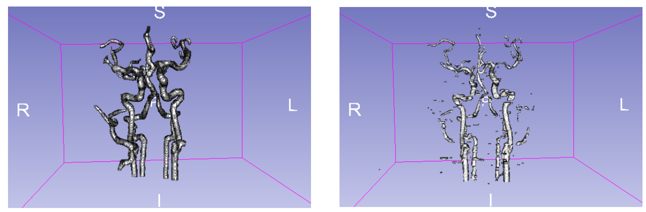
As we can see, adding the drop layers, doing some data augmentation & combining the two U-Nets architectures improved the performances of our model (based on the IoU metric). The effect is non-negligible for the 2D models, with a IoU improvement of 0.09 (28.1% improvement). For the 3D models, we can first see that this improvement is less impressive (only a IoU improvement of 0.04≡11.4%). It is also important to note that the best 3D model has a worth validation IoU than the best 2D model.

3.4. Discussion

First, we can see that the models indeed learn to segment the arteries while minimizing the loss function (*i.e.* the Dice coefficient), the IoU metric value increase as expected. In addition, the training IoU reaches high values, above 0.7, meaning that this arteries' segmentation task is suited for these kinds of models. However, validation IoU can't reach these high values since the training dataset is relatively small with only 4 images.

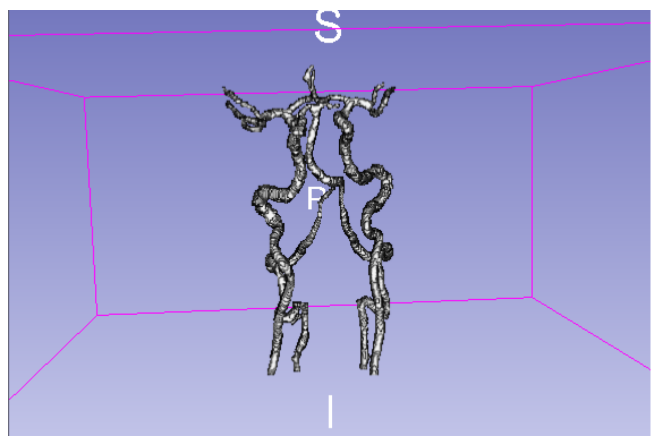
Another interesting point is the fact that 3D models seem to give worse results than 2D models. One possible explanation is the fact that when going from 2D models to 3D models, we increase the number of trainable parameters (from $\sim$ 2 millions to $\sim$ 7 millions) while the number of different inputs are reduced (since we have to stack several 2D inputs to produce a single 3D input). We therefore have to train a more complex model with less data, knowing that the model already had overfitting issues before.

The more the model and the annotator agree on the segmentation, the higher the IoU score is (with a maximal agreement of 100% ≡ IoU of 1). However, the IoU metric can't **entirely** express which model's prediction fulfill the best the doctors needs. Therefore, full predictions have been made (using the validation CT scan) both with 2D (see fig. 7) and 3D (see fig. 8) models.

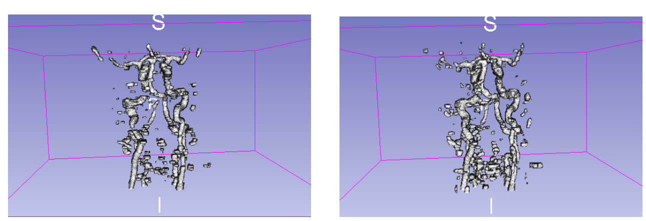


(a) true mask (b) prediction using 2D model

Fig. 7. Best 2D model prediction.



(a) true mask



(b) prediction using Z=2 3D model (c) prediction using Z=4 3D model

Fig. 8. Best 3D models (with Z=2 & Z=4) prediction.

4. CONCLUSION

The results show that U-Net models are suited for this cerebral arteries' segmentation task. Their optimization seems promising. However, with so few images, the validation IoU reached a plateau close to 0.4. It would be interesting to know if, with more training data, the validation IoU could reach the same level as the training IoU which exceeded 0.7.

Two interesting applications of such models could be for cerebral arteries' segmentation without contrast agency, reducing the side effects of these iodized products or for strokes detection and fast visualization.

Finally, one may argue that U-Net does not take into account the spacial continuity of arteries. It would then be interesting to explore other machine learning models that could take into account this continuity (such as CNN for time series).

$Note$: The code used in this project is available on GitHub: https://github.com/LorisPilotto/ArteriesSegmentation

5. REFERENCES

[1] Chen Riang's Blog: F1 Score = Dice Coefficient, https://chenriang.me/2020/05/05/f1-equal-dice-coefficient/, 2020

[2] Wikipedia: U-Net, https://en.wikipedia.org/wiki/U-Net, 2021

[3] zhixuhao: Implementation of deep learning framework – Unet, using Keras, https://github.com/zhixuhao/unet, 2018