Cascaded U-net++ for segmentation of lung lesions

Nathan Habib
Supervisor: Nicolas Boutry

July 6, 2021

1 Introduction

Being able to segment lung lesions from chest CT-scans is a crucial step in the automated diagnosis of patients with lung disease and the evaluation of the latter's progression. Also, assisting medical staff has become one of the priorities of recent AI development, however, the medical field is one that cannot permit blind algorithms to give their diagnosis without us knowing what they are actually doing, explanatory AI thus become a priority in the development of every new algorithm. Furthermore, to be able to assist medical staff (or anyone really) the AI has to be accessible, clear and easy to use, the best AI in the world is useless if no one can use it. We therefore experimented in the development of an interface between a user (medical staff) and the algorithm that is going to be doing the segmentation.

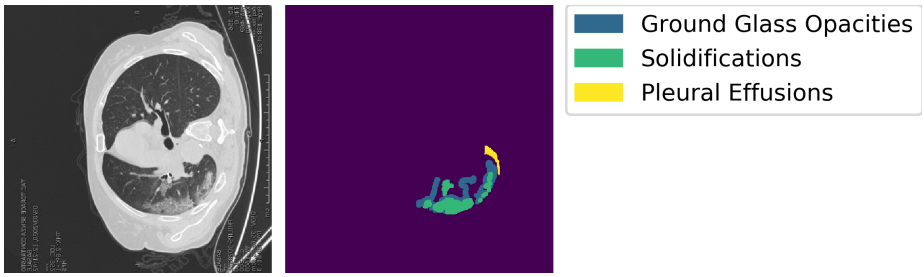
2 Motivations and Goals

At first, this project's goal was to participate in the COVID-19 Grand Challenge [1]. The goal of this challenge is to diagnose a patient that might be infected with Covid-19, using a chest CT-scan of that one. The challenge's organizers proposed a method to tackle the problem [3]. This method proved to be difficult to recreate properly, and we were met with poor results. We therefore decided to first segment out the lung and the different type of lesions one by one, and, from there, evaluate the probability that the patient is infected. After consideration, we eventually gave-up on the challenge. The goal then became the segmentation of the lung and of multiple lesions, for now, Ground Glass Opacities, Solidification and Pleural Effusions, and then do a diagnosis on the patient.

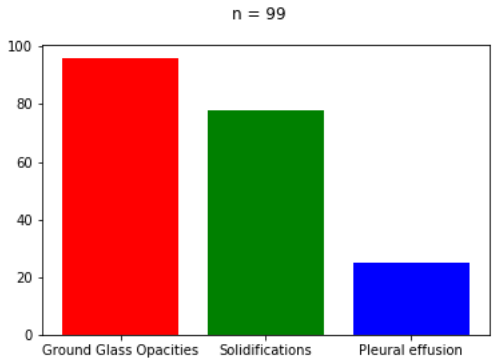
3 Evolution of Methods

3.1 Data

The data has been gathered from a single source [4]. This source contains two different datasets, one consisting of 99 chest CT-scans with annotations containing ground glass opacities, solidification and pleural effusions' (Fig. 1). The other one consisting of 450 chest CT-scans, annoted to include lungs, Ground Glass Opacities, and solidification (Fig. 2).

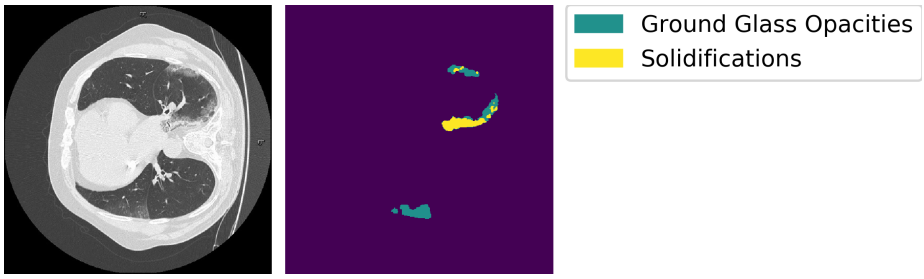


(a) random sample



(b) lesions distribution

Figure 1: dataset 1 description

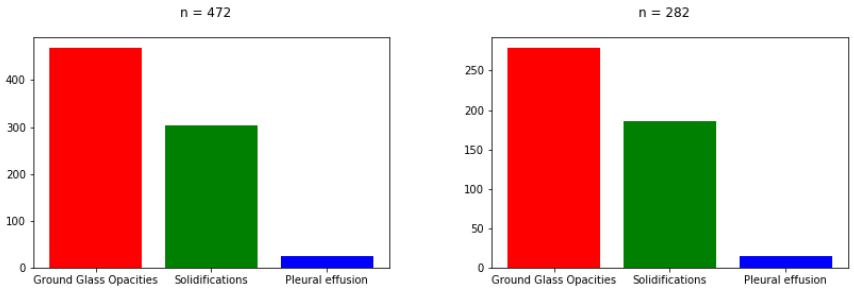


(a) random sample

(b) lesions distribution

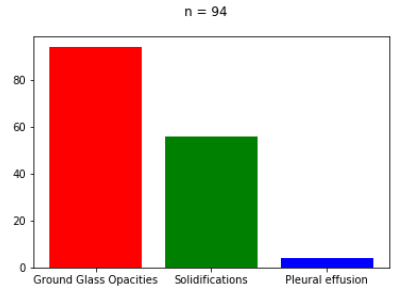
Figure 2: dataset 2 description

To get a maximum of training sample and avarietyin the type of lesion we observe, we concatenate the two datasets. The main downside with this method is that now we have an important imbalance of the type of lesions present in the dataset (Fig. 3).

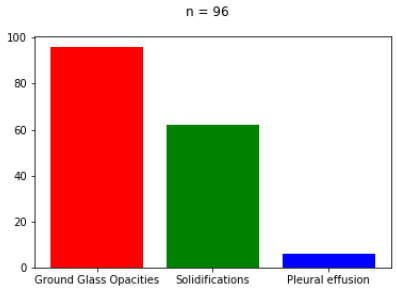


(a) whole dataset

(b) training dataset



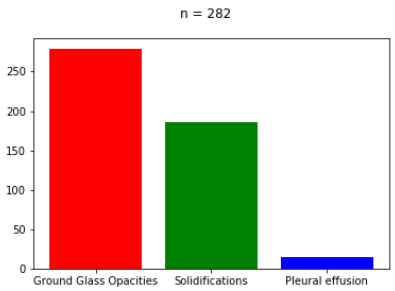
(c) validation dataset



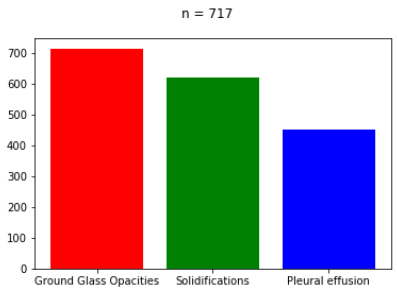
(d) testing dataset

Figure 3: Lesions distribution in mixed datasets before augmentation

To solve this problem we augment the training dataset. The augmentation consists of rotations from -29º to 29º with a step of 2 (Fig. 4).



(a) non augmented



(b) augmented

Figure 4: Lesions distribution in training dataset before and after augmentation

3.2 Model Architecture

The model architecture has undergone a few evolutions, from the loss function to the network structure itself. This section consists of the evolutions of the different methods we tried to arrive at the one we use now. This is useful to better explain the choices we made for our current approach as well as why it works best.

3.2.1 U-net and Multiclass Segmentation

The data at our disposition consist of a chest CT-scan accompagned with one masks on which all the classes are represented at the same time. Our first approach was then the most naive one: a simple multiclass segmentation with a categorical crossentropy loss. (Fig. 6).

We chose the U-net [6] architecture (Fig. 5) because it is already proven to be very effective in medical imaging segmentation [2].

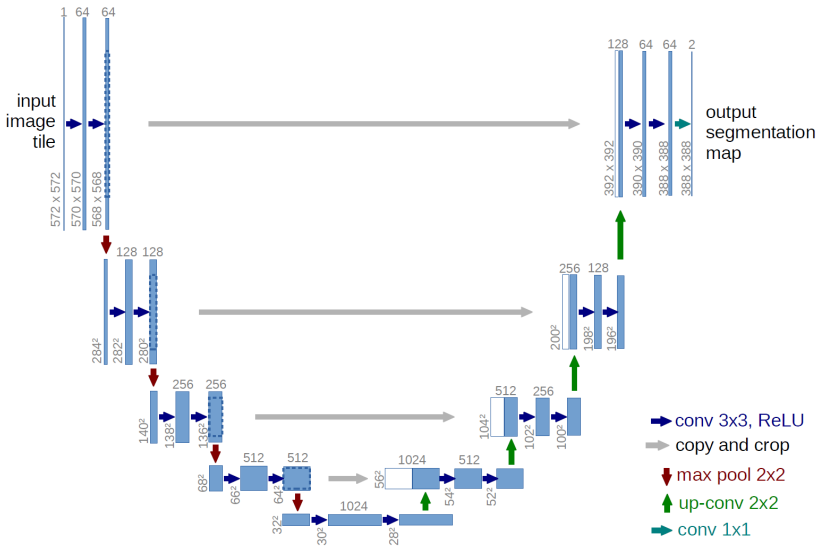


Figure 5: U-net architecture

At this point we experimented with multiple number of the first layer filters (1, 2, 4, 8, 16, 32, 64) (Table. 1) because our network might be too "powerful" to learn with so few examples. We found that 64 worked best in most cases.

layer 1	layer 2	layer 3	layer 4	layer 5
1	2	4	8	16
2	4	8	16	32
4	8	16	32	64
8	16	32	64	128
16	32	64	128	256
32	64	128	256	512
64	128	256	512	1024

Table 1: Number of filters per layer in U-net

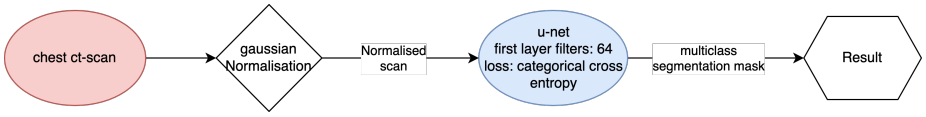
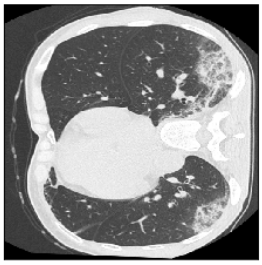
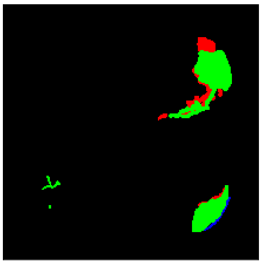


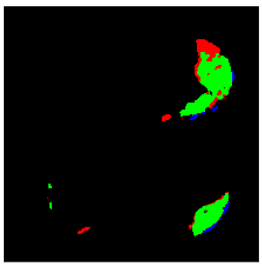
Figure 6: Multiclass Segmentation of lesions



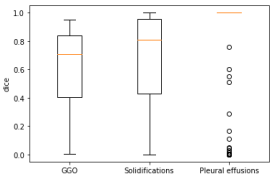
(a) Ct-scan, (input image)



(b) Segmentation Ground Truth



(c) Prediction



(d) Performance

Figure 7: Green: GGO; red: solidifications; blue: pleural effusions.

We can see here that the network has trouble segmenting small regions and differentiate between the different types of these legions.

3.2.2 U-net and Binary Segmentation of Each Class Individually

To solve the segmenting issues mentioned above, we tought of using 3 dedicated networks one for each lesion (Fig. 8).

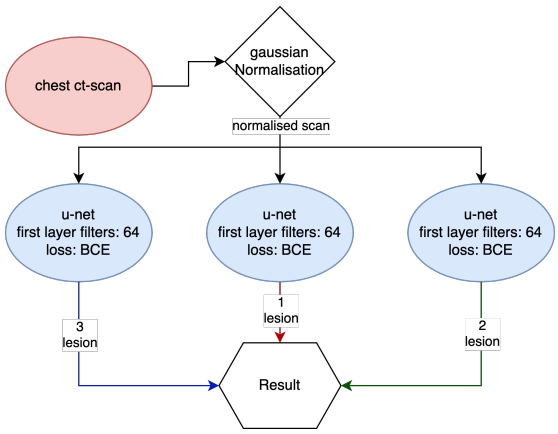
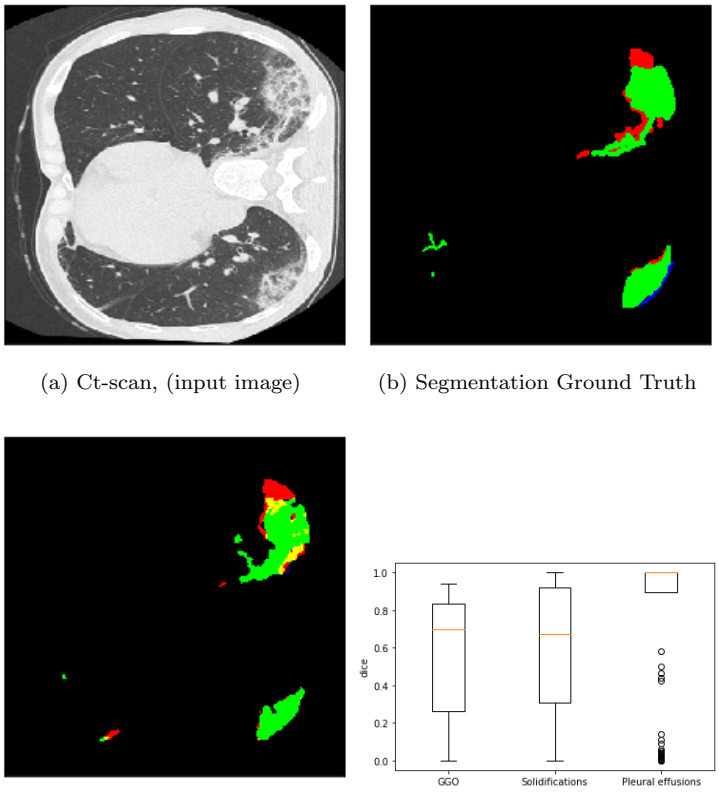


Figure 8: Binary Segmentation of individual lesions



(a) Ct-scan, (input image)

(b) Segmentation Ground Truth

(c) Prediction

(d) Perfomance

Figure 9: Green: GGO; red: solidification; blue: pleural effusions. The yellow lesions are where the networks are in conflict, here the network think that there is both GGO and solidification at the places where we see yellow.

3.2.3 Cascaded U-net and Binary Segmentation of Multiple Classes

The performance of the segmentation of individual classes was not doing better than the multi class model it was even a bit worse. However, we still thought that having dedicated networks for each lesion would lead to superior results so, we kept this idea and try to improve on it. One improvement that came to mind was, segmenting larger regions by adding together the lesions to segment. Thus, to be able to segment out all 3 classes we still only need 3 networks. The first to segment out all classes, the second to segment out two out of the previous 3 and the last one to segment 1 out of the previous 2. To facilitate the work for the network we feed the concatenate the output of the previous network with the input of the one coming after it thus resulting in cascading networks (Fig. 11). This method has been shown to work with brain tumors where the network first segmented the tumor and sub-networks then used that

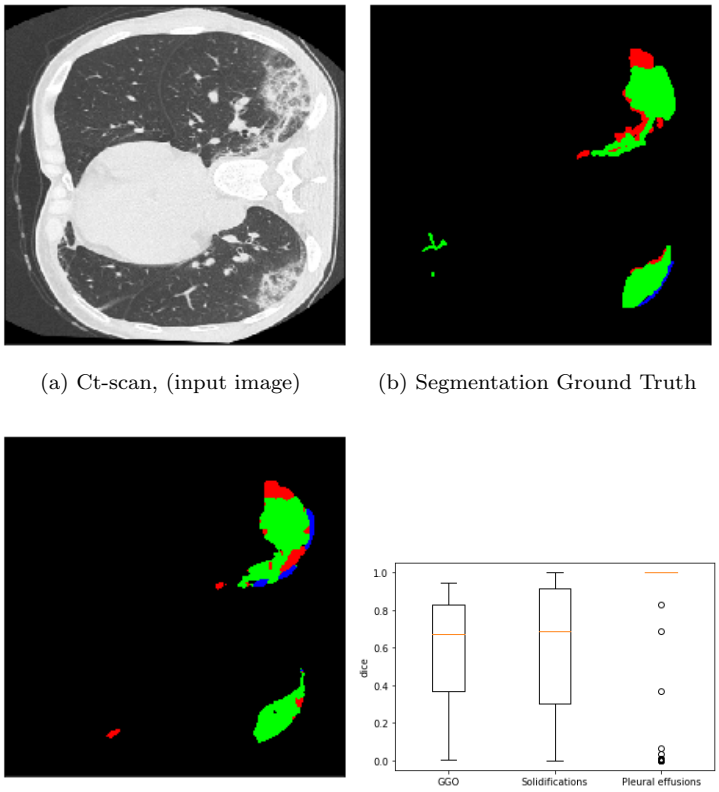
information to segment-out finer and details [9, 10]

The fact that in medical imaging the zone to segment is often minuscule compared to the background makes it so that we often get trap in a local minimum of the binary crossentropy loss function when training, this leads to a network that is strongly biased to recognize background everywhere and so, the zone of interest is often missed by the network. To tackle that problem we need to have a loss function that will be strongly biased towards the zone of interest, (the loss will be equal to 1 if no foreground pixels are recognized even though all background pixels are correctly classified). We therefore introduce the Dice function (equation) which, when transformed to a loss function, satisfies this particularity. However, we found it difficult to make the Dice loss converge, we therefore used a linear combination of the Binary crossentropy and Dice loss [5]:

$$\mathcal{L}(Y, \hat{Y}) = -\frac{1}{N} \sum_{b=1}^{N} \left(\frac{1}{2} \cdot Y_b \cdot \log \hat{Y}_b + \frac{2 \cdot Y_b \cdot \hat{Y}_b}{Y_b + \hat{Y}_b} \right) \tag{1}$$

Where Y is the ground thruth, $\hat{Y}$ the output of the network, both of the b^{th} image and N is the batch size.

Another option was to use use structural similarity loss, again in conjunction with BCE, however this led to poor results.



(a) Ct-scan, (input image)

(b) Segmentation Ground Truth

(c) Prediction

(d) Perfomance

Figure 10: Green: GGO; red: solidifications; blue: pleural effusions.

We can see from the dices that this model does indeed better than the previous one.

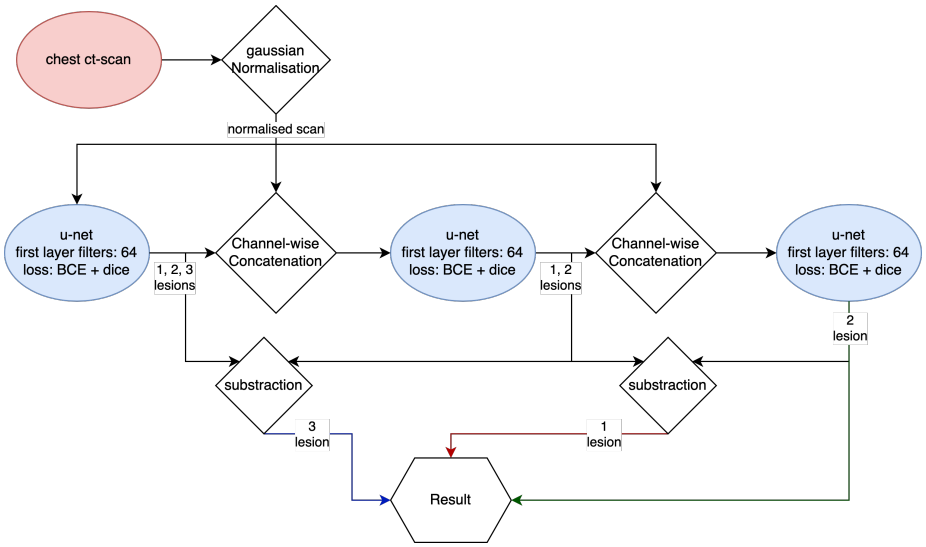


Figure 11: Cascading U-nets

3.2.4 Cascaded U-net++

During our research we found a variant of the U-net architecture more efficient towards segmenting lesions that varies greatly in size, the U-net++ architecture [12]. We found that the objects we are working with (lung lesions) tend to vary greatly in both size and shape we therefore implemented this architecture. (Fig. 13)

The U-net++ architecture differs from the classic U-net in 2 ways:
First the introduction of dense blocks in as a remplacement for the skip connections.
Second, the possibility to use as output the mean of the four intermediate convolutions of the first layer (Fig. 12). We will call that the accurate mode, in contrast with the classic mode (use only one output) that we will refer as speed mode (this is the nomenclature used by the original paper) [12].

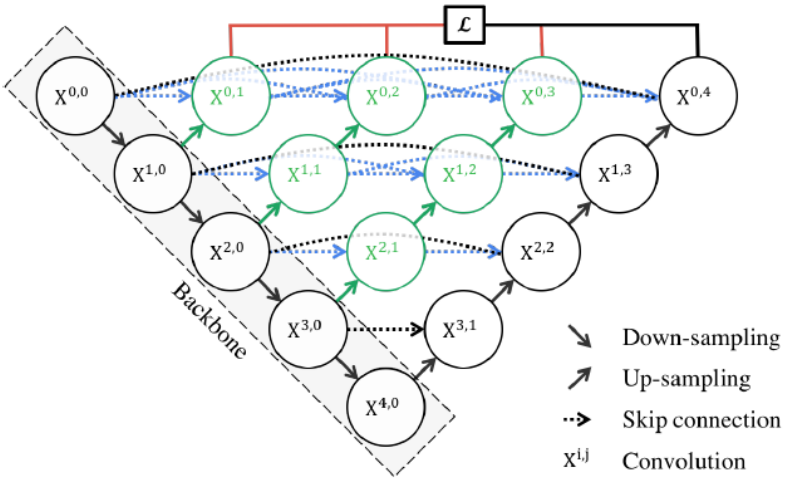


Figure 12: U-net++ architecture

This helps getting better results for lesions varying greatly in size because the receptive field of each intermediate convolution is different and thus will be more prone to pick up differently sized regions.
We therefore used the accurate mode to segment the lesions and normal mode to segment the lung. We do not use the accurate mode for the lungs because the do not vary in size (or only in small amount) and we will then see a performance drop [12].

The addition on convolutions in the skip connections make it so that the number of parameters is much larger than a classic U-net for the same number of first layer filters. Thus we only use 32 filters in the first layer of the U-net++ architecture. Using 64 filter in the first layer make it so that there is 36 millions parameters, this only around 2 millions more than a classic U-net however, the number of connection between those layers is much bigger, so big that the training becomes difficult (or impossible) on a single GPU.

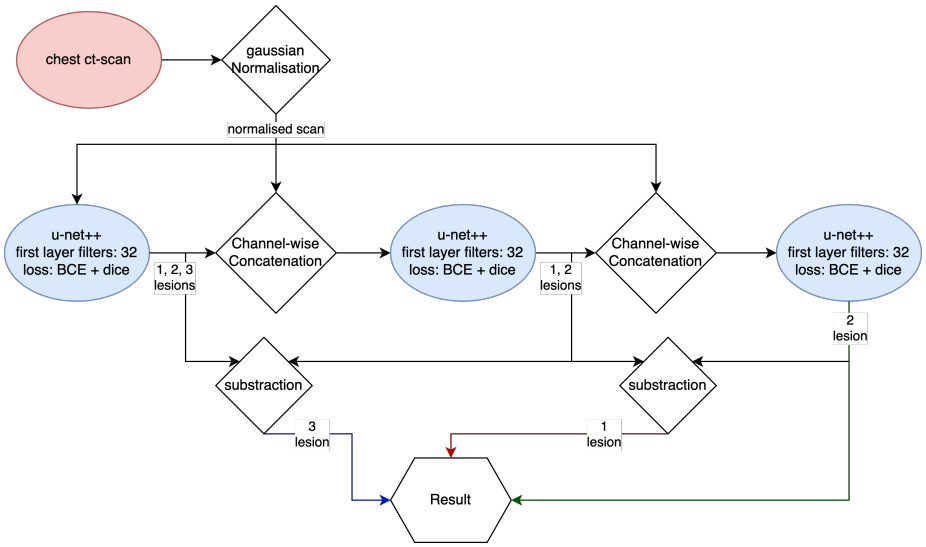
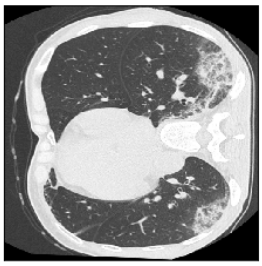
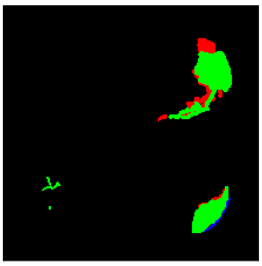
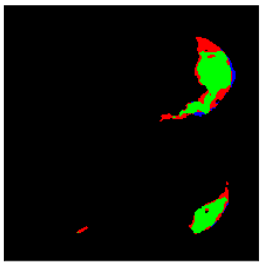


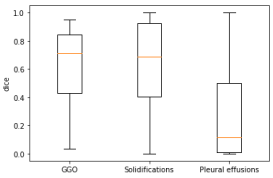
Figure 13: Cascading U-nets++





(a) Ct-scan, (input image) (b) Segmentation Ground Truth





(c) Prediction (d) Perfomance

Figure 14: Green: GGO; red: solidifications; blue: pleural effusions.

3.2.5 Comparison of results and limitations

To our surprise, the multiclass model is generaly at least as accurate as the cascaded U-net++ model. Our theory is that the innacuracies of the cascaded U-net++ comes from the cascade itself. Indeed, innacuracies from the first layers propagate and get amplified the deeper we travel in the cascade. We could therefore be led to believe that a multiclass model using the U-net++ architecture would be the most efficient.

Even though we tried our best to reduce biases in the data and in our methods, there still remains many components limiting our findings.

There is in a first time limitations concerning our data. The data we use come from only one source, it consists of overall good quality images, of sick patients. This poses a problem because the model is biased towards findings lesions (even if there is none) and performs very poorly on lower quality images (maybe even on better quality because the informations present will not be the same as during the training). Furthermore, we only are able to segment 3 lesions type, again this poses a problem in real world use. Finally CT-scans usualy come in slices of one patients, representing 3D informations, our model only looks at one slice at a time and does not take into account other slices of that same patient. To tackle that problem we could use a 3D segmenting architure however this make it impossible to analise only one slice of a patient, (if only one or a few slices are available we cannot use a 3D U-net), thus maybe a recurent model could solve that issue, taking in as many slices of a same patient and taking every one of them into account for the segmentation.

Then we have limits concerning the comparaison of the models.

First, we did not have the time to train all the models multiple times, only the multiclass and cascaded U-net++ have been trained 3 times, and average, however, we found that the results for all the trainings dones we varying only slightly. Furthermore, the testing sets were unbalanced lesions-wise, which makes the results, for the pleural effusions not representative of real performance.

3.2.6 Gaussian normalisation

We normalize all images that goes into the models using a Gaussian normalisation:

$$gaussNorm(X) = \frac{X - mean(X)}{stdDev(X)} \tag{2}$$

3.2.7 Census Transform

To recognise lungs in a picture we do not need to see the any texture variation, the only relevant thing to look for is the structure of the lung moreover in CT-scan, different patients can have different lung tissue density and as a consequence the value intensity of the zone of interest (the lung) will change. We could easely recognise the lungs in a binary image in contrast with the lesions for which we need to identify texture differences inside the lung. Thus to eliminate texture variation in different CT-scans and highlight the structures of the different element present in the image we use a variation of the Census transform [11] (Fig. 15).

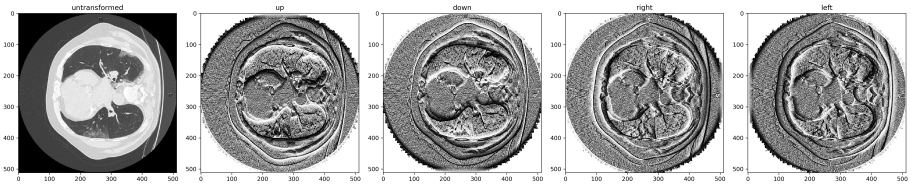


Figure 15: Census transform with range = 15

4 Explainable AI

One of the main challenge of AI today is being able to explain the decisions that it takes. In fact, being able to see more clearly into these algorithms is a prerquisite for it to be available in a critical environment, one cannot trust an AI with the life of a patient if we are unable to understand the inner workings of that AI, the same goes for self-driving and countless other applications.
With this in mind, we experimented with multiple approaches trying to explain the decisions of our four networks and the differences in their performances. This has proven to be difficult as explainable AI (xAI) is a fairly recent field of research and not much literature has been found on the subject of xAI for image segmentation.
We first tried using the features maps of each networks to see what was activating the network, and have a first impression of the zones of interests. However this is a very poor insight into the network's inner workings.
We therefore thought about applying a method used for classifications tasks called Integrated Gradients (cite). However, this method proved to be difficult to implement for image segmentation.

4.1 Features Maps

The features map of a network are the outputs of the convolutional layers inside it. By looking at those activations we hope to determine what area of the image

"excites" or "inhib" the network the most, we should be able to detect the outline of the lung and hopefully the textures of the lesions as they are being percepted by the network.

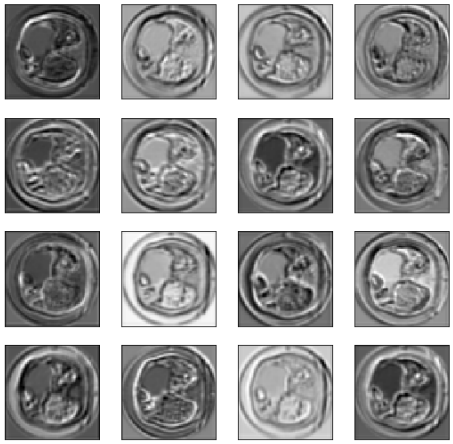


Figure 16: 16 feature maps (randomly selected out of 512)of the output of the fourth layer

We show here the features map at the output of the fourth encoding layer because it is the one where we can most clearly see the activations of point of interest (Fig. 16). The results are as expected, we can see the perimeter of the lung as well as the lesions being picked up by the convolutions in each layer.
We do not believe this method to be useful in making sense of the network's choice because of the broad interpretability of the feature maps, the deeper we go in the model, the more abstract the activations become.
This observation led us to try the comming method.

4.2 Integrated Gradients

Integrated gradients is a method consisting of attributing an certain level of importance in the decision of a network to its input. It is commonly use in image classification, where we attribute an importance in the decision of a particular class, to each pixels of an image [7].

$$IntegratedGrads_i^{approx}(x) := (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m} \quad (3)$$

Where i is a feature (an idividual pixel in our case), x is the input (image tensor), x' is the baseline (all zeros image tensor in our case), k is a scaled feature pertubation constant, m is the number of steps in the rieman approximation of

the integral.

It can also be used in sentiment analysis and other classification tasks. However to the best of our knowledge this technic has rarely been used in the context of image segmentation, and we are the first to try to implement the integrated Gradients algorihtm for image segmentation. One paper did to provide visual explaination for semantic image segmentation but using Grad-CAM another gradient based algorithm for that purpose [8].

Implementation of Integrated gradients for image segmentation has proven to be difficult. We only concentrated on binary segmentation in a first time. The process was as follow: first take a positively classified pixel and apply the integrated gradients technic on it, that is, highlight what pixels in the input image contributed to the classification of this pixel as zone of interest.

Then we tried to get the integrated gradient to every positively classified pixels in the image and show the results in as one heatmap to show what parts of the image was responsible to the classification of the whole zone of interest, we hoped to see clear structures to be highlighted, for example, the perimeter of the lung, which would indicate that the network knows that a lesion can only be found inside a lung.
However, our images being 512px by 512px the number of correctly classified pixels can explode rapidely, (as much as 20,000). This made the algorithm very slow and we are still in the process of finding a solution for that problem, (find a way of getting the gradients for all the pixels as fast as possible). In the meanwhile we ran the integrated gradients algorithm with only a few number of steps and by skiping one in every 5 pixels (see explication of Integrated gradients) and got those results:(Fig. 17). Those results show us something very interesting, the model is not looing at the context in which the lesions finds himself, it is only looking at the texture changes in the image to determine if the pixel is a lesions or not. This is obviously a huge problem which could be solve by giving it images with similar texture difference but without the lung and tell the network that it is not a lesion. Hopefuly this will be enoug hfor the model to learn about context.

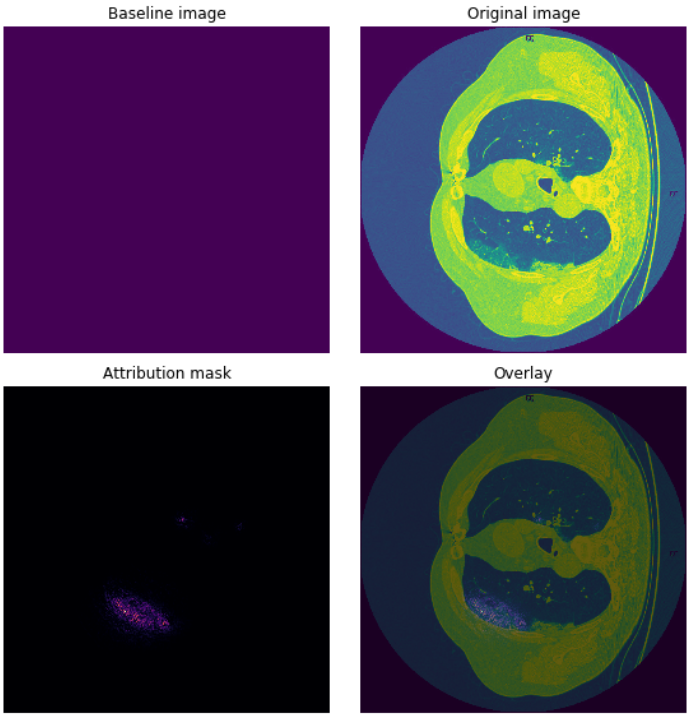


Figure 17: Integrated Gradients for 1 in every 5 lesions pixels

5 Interface

For people to be able to benefit from our research it is of most importance that it presents itself in the simplest manner possible. That is why we found it important to develop a web interface to use the segmentation algorithms. This interface only has two buttons, a button to load the image we want to segment and one to segment said image. Once we click on the segment image we are presented with 3 images: the original image, the image with the segmented lungs highlighted and the images with the different type of lesions highlighted in different colors (a legend is found next to the images to discern between the lesions). Then, we also find a percentage of lung coverage for each lesions (Fig. 18).

6 Conclusion

This project had for goal to investigate methods for efficient and accurate segmentation in lesions in chest CT-scans, developping a sense of what the AI was doing "under the hood" and presenting an interface for a potential end user.

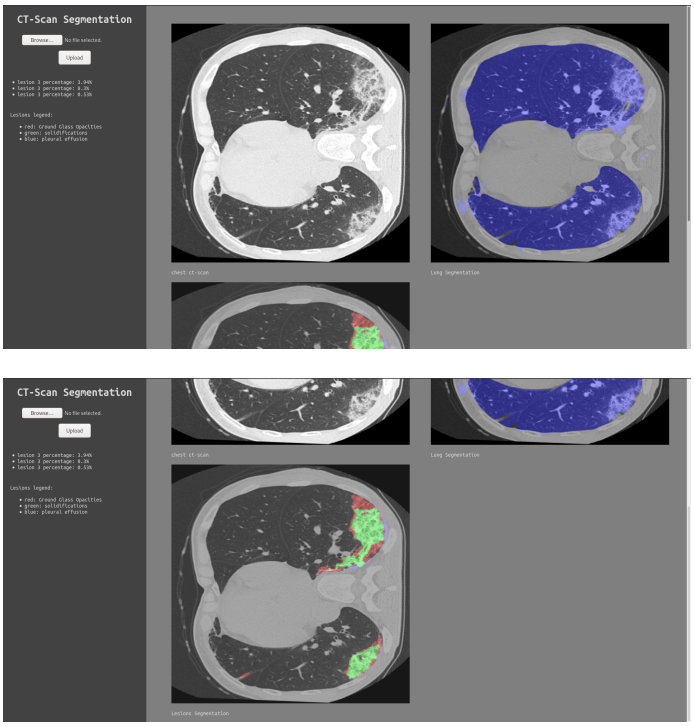


Figure 18: screenshots of the application, the second image for the same CT-scan as the first it is just scrolled down

We will now do a roundup of those three objectives, what has been done and what is left to explore.
The most important objective was to find a way to segment out the lesions in a chest CT-scan, for use in diagnosis of patients, we found that the state of the art, even if performant, was not accurate enough for reliable diagnosis, we thus tried multiple method involving cascaded models and changes in the architecture of the models (U-net++), loss function (Dice loss) and preprocessing (Census). Even if promising we do not think that these method make for a model efficient enough for the real world (again it might be a big lack in the diversity of the data). Some exploration paths would be implementing the U-net++ for multiclass segmentations, and a recurent neural network to take advantage of the relationship between the many slices of a patient (without the bulk and hassle of a 3D model).
During the development of a model an immensly important step was the comprehension of that model's inner working, we wanted to find a way for it to explain itself when it made a decision and have a sense of why it was more efficient. In that regard we managed to roughly implement the integrated gradients algorithm for image segmentation (though there still is a lot of work to do) that allowed us to know that the model is solely looking at the texture difference in

the image (without context of the lung).
Finally, the user interface is usable and runs relatively fast on a dedicated GPU (a few seconds for one image with the cascaded U-net++) however it is to be greatly improved by adding the possibilty to upload nibib images, download the results (images and lesions percentage) and upload multiple slices at the same time (for comparaison).

References

[1] Covid grand challenge 2020. https://covid-ct.grand-challenge.org/Data/. Accessed: 2020-07-13.

[2] Intisar Rizwan I Haque and Jeremiah Neubert. Deep learning approaches to biomedical image segmentation. *Informatics in Medicine Unlocked*, 18:100297, 2020.

[3] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *medRxiv*, 2020.

[4] MedSeg. annotated chest ct-scans, 2020. data retrieved from the medseg website: https://medicalsegmentation.com/covid19/.

[5] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[8] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. Towards interpretable semantic segmentation via gradient-weighted class activation mapping. *arXiv preprint arXiv:2002.11434*, 2020.

[9] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks. In *International MICCAI brainlesion workshop*, pages 178–190. Springer, 2017.

[10] Guotai Wang, Wenqi Li, Tom Vercauteren, and Sebastien Ourselin. Automatic brain tumor segmentation based on cascaded convolutional neural

networks with uncertainty estimation. *Frontiers in computational neuroscience*, 13:56, 2019.

[11] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994.

[12] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018.