Internship Report

Coronary Artery Calcium Scoring from Non-Contrast Cardiac CT Using Segmentation and Deep Learning

Student :
Julien Triacca
(julien.triacca@gmail.com)

Supervisor :
Hyunho Mo
Esther Bron

20 septembre 2024

Summary

1 Introduction

Cardiovascular diseases (CVD) refer to all pathologies affecting the heart and blood vessels. CVD and their complications are the leading cause of death worldwide, with 17.9 million deaths in 2019. It is also the leading cause of premature death. By 2030, an increase of +3% in all cardiovascular diseases and +8% in strokes is expected. The stakes in terms of screening, prevention, and support are crucial both in health and socio-economic terms. The resources deployed in research are evolving.

This is where the Coronary Artery Calcium (CAC) score comes in, which is part of the risk stratification strategy for CVD, particularly in asymptomatic or minimally symptomatic patients. It allows for the detection and prevention of cardiovascular risk in a patient. It corresponds to a numerical evaluation of the extent of calcified atheromatous deposits observed in the walls of the heart's arteries, the coronary arteries. The CAC score now occupies an important place in cardiovascular risk stratification.

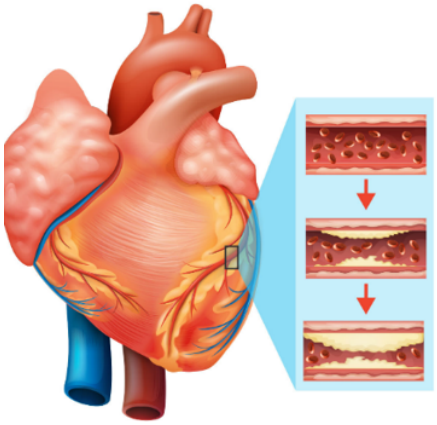

FIGURE 1 – Drawing of Calcium inside the arteries

In this project, my goal is to work on a part of Hyunho Mo's project, which aims to associate a CAC score with a patient. For this, we notably use the Agatston score, which will then be added to tabular data such as age, sex, medical history, etc., in order to assign a risk of cardiovascular accident. The method used to evaluate the CAC score involved several stages of segmentation, as shown in Figure 2 below.

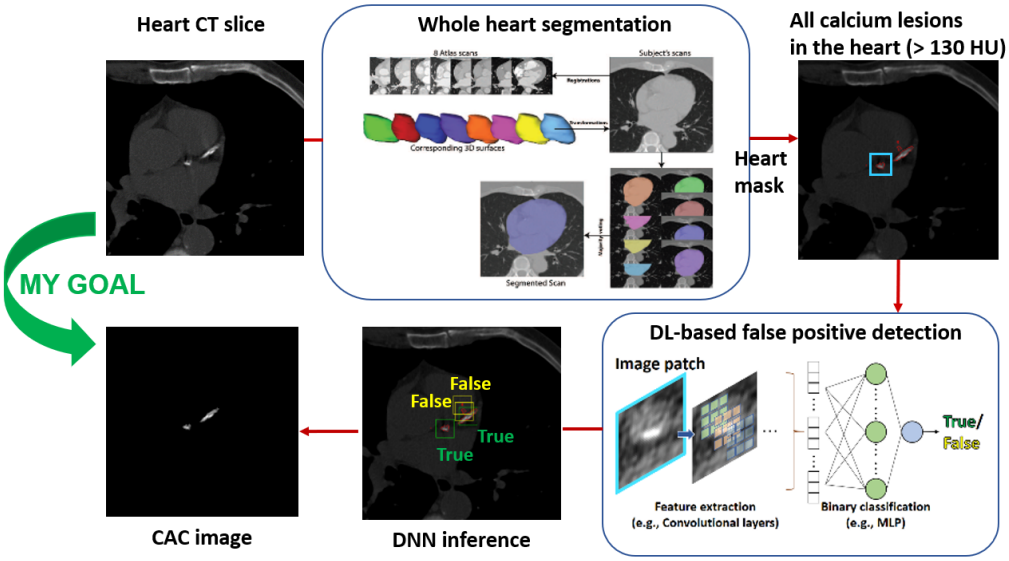

FIGURE 2 – Initial method to evaluate the CAC score

My objective here was to find a method that allows for a direct transition from the Heart CT Slice stage to the CAC Image stage as indicated by the green arrow, in order to simplify the steps and improve accuracy. For this purpose, I focused on a machine learning model known in the medical field for segmentation called nnU-Net. I will explain how I applied this method, the challenges I faced, and the results.

2 The nnUNet method

nnU-Net is a versatile and self-configuring semantic segmentation framework designed for biomedical image analysis. nnU-Net has proven its effectiveness in numerous competitions, achieving top positions across various datasets in the biomedical domain. It supports 2D and 3D images across different modalities and handles varying image sizes and class imbalances with ease. The method relies on supervised learning, requiring training data but often less than other solutions thanks to its robust data augmentation strategies. nnUNet is ideal for both domain scientists seeking to analyze medical images and AI researchers looking to develop or test new segmentation methods, providing a strong baseline and a method development framework. It's optimized to process entire images in one go during preprocessing and postprocessing phases, accommodating a wide range of image sizes as long as hardware resources permit. This is a funnel model with a first part as an encoder and a second part as a decoder

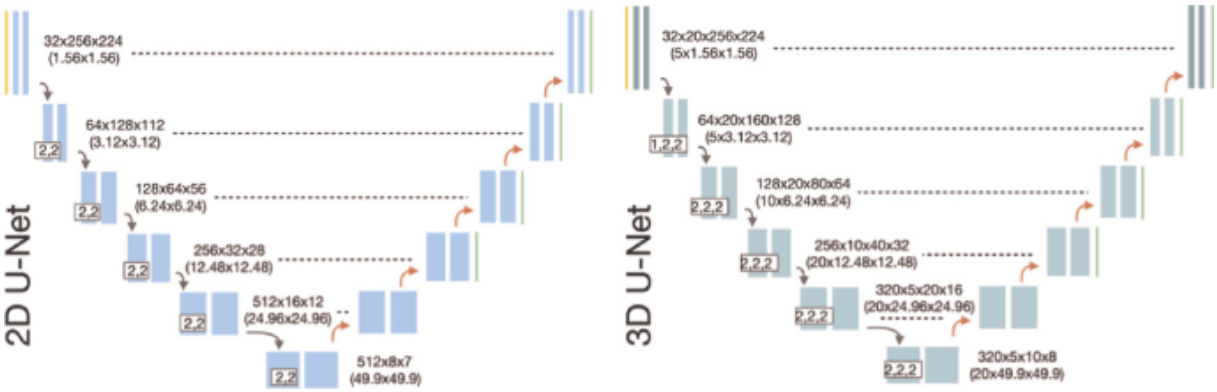

FIGURE 3 – Initial method to evaluate the CAC score

nnUNet is widely used in the medical field for segmentation, offering numerous advantages, including the ability to support both 2D and 3D inputs. It is available as open source on GitHub.

3 Database COCA - Coronary Calcium and chest CT's

To be able to train models, the limiting factor is very often the database. Indeed, it is complicated to have a complete database, and databases often have restricted access (cost or security). However, in the context of non-profit scientific research, I was able to use the COCA database provided by Stanford. This database comprises two different components : non-gated chest CT DICOM images with coronary artery calcium scores, and gated coronary CT DICOM images with corresponding coronary artery calcium segmentations and scores (xml files) ; it is the latter that we will use.
We thus have a database for 450 patients, each with a file corresponding to the CT scan of the heart, accompanied by an annotation. Regarding the CT scans, these are DICOM files, so images accompanied by metadata such as the real scale, etc., which will be useful later. Each patient has about 50 CT scans corresponding to slices spaced 3 mm apart. As for the annotations, these are XML files corresponding to each of the patients ; we have the coordinates of the contours of the areas where calcium has been detected. This will

be used for subsequent supervised learning. We have the information on which of the four main arteries this corresponds to, but this will not be useful later on.
In order to visualize these DICOM files and better understand the database, I used the software XnView.

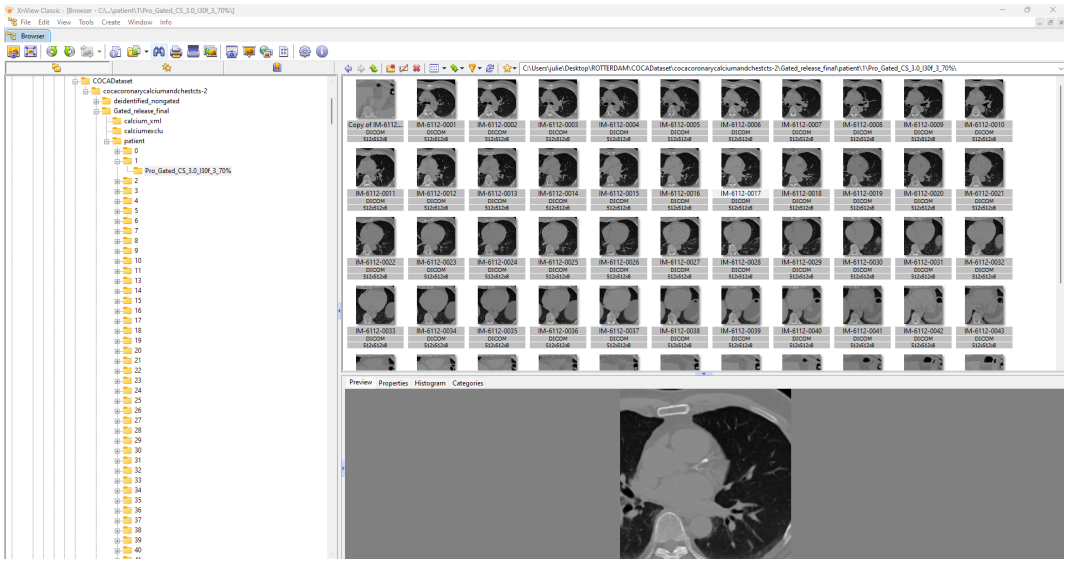

FIGURE 4 – Preview of our CT scans using the XnView software

This database proved to be very useful because it was already annotated, and having to annotate a database would have required a lot of work that I would not have been capable of doing. However, the goal will subsequently be to apply our trained model to other databases to estimate its effectiveness and generalization.

4 Database Preprocessing

Regarding the database, although it was somewhat organized, the data was neither sorted nor structured as needed. Therefore, I had to write a script to preprocess each file to ensure it conformed to the input requirements of my training nnUNet, both in terms of structure and nomenclature.

Additionally, the data format posed a challenge. As mentioned earlier, my CT scans were DICOM files, which are not accepted as input for nnUNet. Hence, it was necessary to convert them into Nifti format, which is widely used in medical imaging. This is where the first distinctions between the 2D and 3D versions arise, which I will detail later.

I developed a script to convert the annotation coordinates into masks in Nifti format, ensuring consistency with the CT scans. Subsequently, these had to be superimposed to verify correspondence and ensure there were no index errors.

During this preprocessing stage, some subjects had to be excluded from our study due to anomalies that prevented their processing. Specifically, 5 out of the initial 450 subjects were excluded.

It was also necessary to split our database into training and test sets. The separation was done with 87 patients designated for the test set, which were set aside and not used until after the training, and 348 subjects for the training set.

Thus, my data was organized as follows :

— **ImagesTr** : Corresponding to the training CT scans
— **LabelsTr** : Corresponding to the training masks
— **ImagesTs** : Corresponding to the test CT scans
— **LabelsTs** : Corresponding to the test masks

```
COCADataset/
├── deidentified_nongated
├── Gated_release_final
└── nn-Unetdatabase/
    ├── nnUNet_preprocessed
    ├── nnUNet_raw/
    │   ├── Dataset001_COCA2D/
    │   │   ├── imagesTr/
    │   │   │   ├── CT_001_0000.nii.gz
    │   │   │   └── ...
    │   │   ├── imagesTs
    │   │   ├── labelsTr/
    │   │   │   ├── Mask_001_0000.nii.gz
    │   │   │   └── ...
    │   │   ├── labelsTs
    │   │   └── dataset.json
    │   ├── Dataset002_COCA3D
    │   └── ...
    └── nnUNet_results
```

FIGURE 5 – Simplified overview of the preprocessed Database

As seen above, it is also necessary to have a `database.json` that contains all the information about our database, including the number of elements, their format, and their size.

4.1 Version 2D

For the 2D version, I do not differentiate between patients, meaning that the CT scans are all considered independently. Only at the end, during scoring, do I group them by patient ID. This represents a slight loss of information. Additionally, another piece of spatial information is lost because we lose information about the slices above and below, which is important since they are often correlated. If there is calcium in one place, there is a chance that there is also calcium 3mm above or 3mm below.

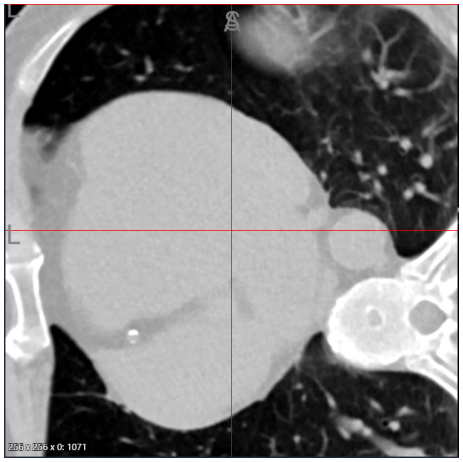

FIGURE 6 – Exemple of CT scan from `ImagesTr`

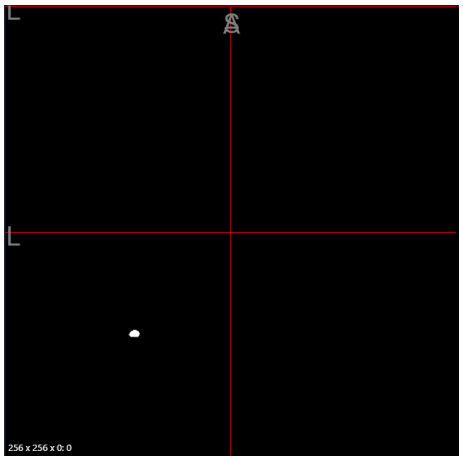

FIGURE 7 – Associated mask from `LabelsTr`

4.2 Version 3D

For the 3D version, the approach is different because the entire CT scan is exported into a single Nifti file. This allows for the preservation of the slice order, thus retaining additional spatial information. It also enables the consideration of a patient's scan in its entirety

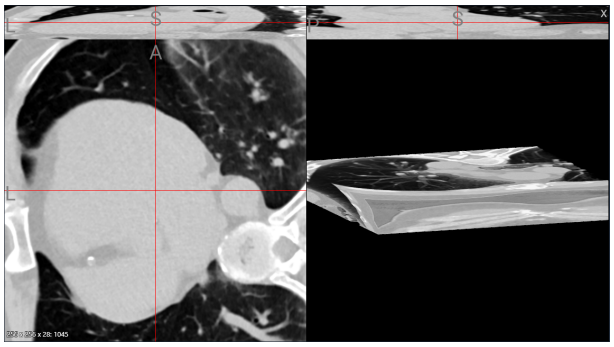

FIGURE 8 – Example of CT scan from `ImagesTr`

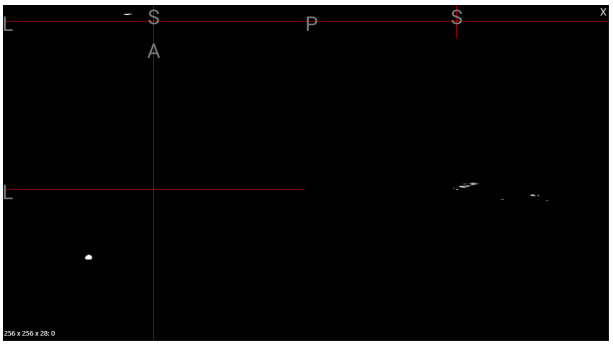

FIGURE 9 – Associated mask from `LabelsTr`

5 GPU Cluster

6 Training

Once our database is organized and formatted to be used as input for our nnUNet model, the training is performed on 5 different folders, each with different validation entries. This allows for cross-validation and enables us to select the parameters from the most performant training

6.1 2D Version

For the training, 200 epochs were sufficient, as increasing the number of epochs showed almost no further improvement in performance. It is also observed that the training and validation loss decrease together, indicating no apparent over fitting, which is a good sign.

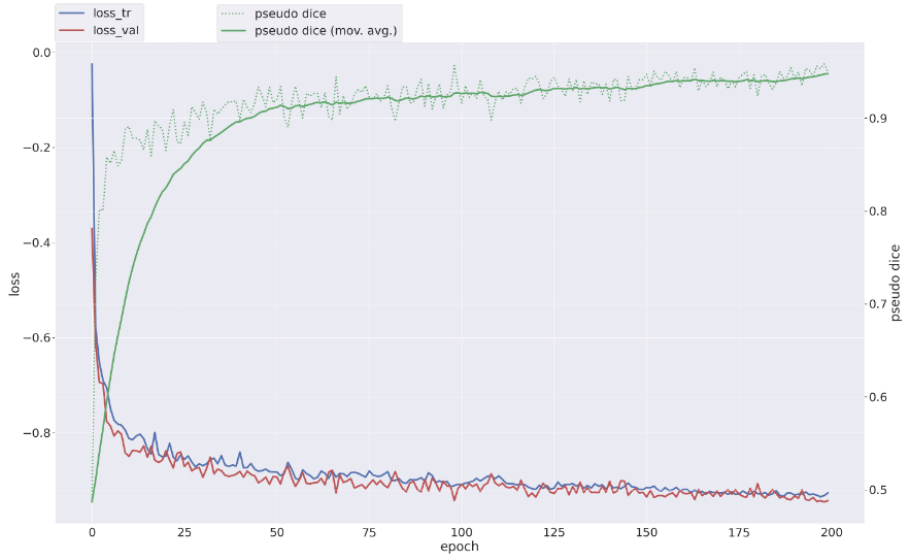

FIGURE 10 – Learning curve as a function of epochs for the 2D version

The duration of the epochs was relatively consistent throughout the training across all different folders. Each folder required approximately 62,000 seconds for training, which is about 18 hours. Fortunately, thanks to the GPU cluster, it was possible to train all 5 folders simultaneously, saving a significant amount of time.

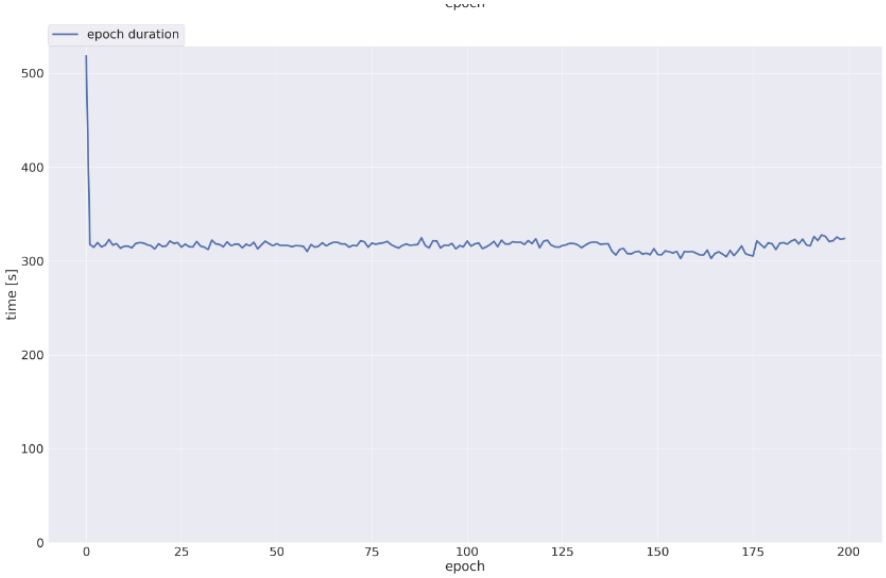

FIGURE 11 – Learning time per epoch as a function of epochs for the 2D version

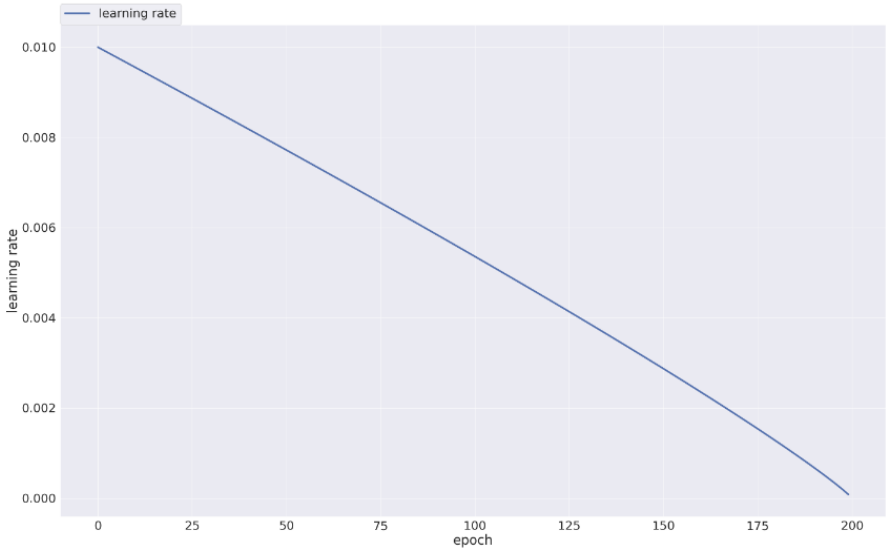

FIGURE 12 – Evolution of the learning rate during training for the 2D version

6.2 3D Version

As with the 2D version, we used 200 epochs. A few more epochs could have improved the results, but restarting the training would have taken too much time, and the improvement would have been minimal. It is also observed that the learning process works well and there is no over fitting.

FIGURE 13 – Learning curve as a function of epochs for the 3D version

Here, the training was much longer than for the 2D version due to the larger size of the inputs. The training duration for one folder was approximately 86,000 seconds, or around 24 hours per folder. Due to the memory usage required by the size of the inputs, it was not possible to train more than 2 folders at a time on the cluster. As a result, the training process was very laborious

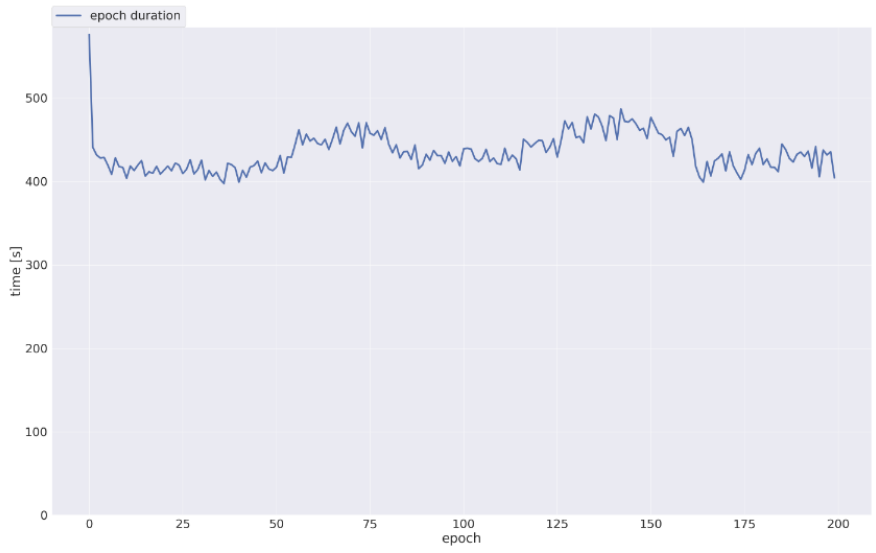

FIGURE 14 – Learning time per epoch as a function of epochs for the 3D version

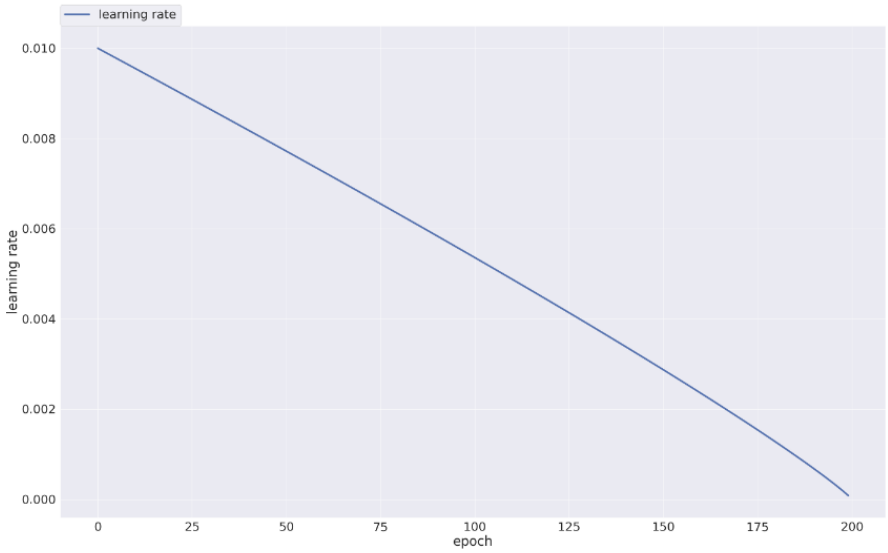

FIGURE 15 – Evolution of the learning rate during training for the 3D version

7 Scoring and Metrics

7.1 2D Version

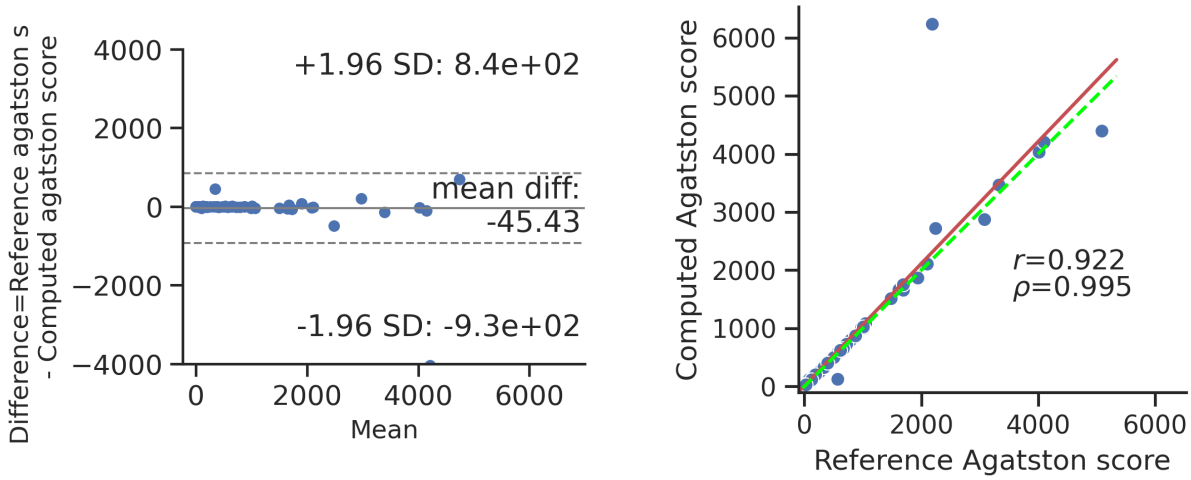

FIGURE 16 – Comparison of the given Agatston score and the derived Agatston score for the 2D version

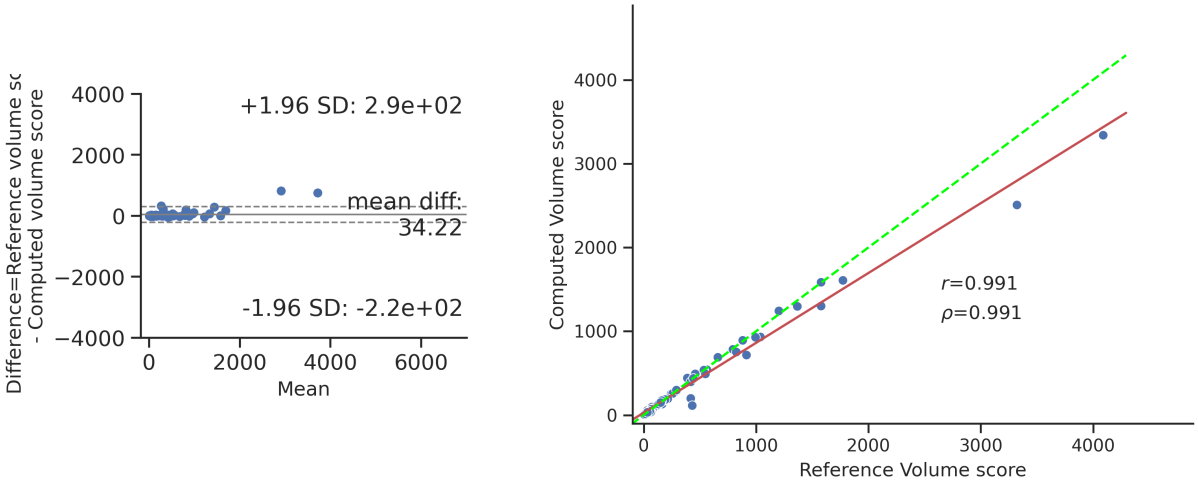

FIGURE 17 – Comparison of the given Volume score and the derived Volume score for the 2D version

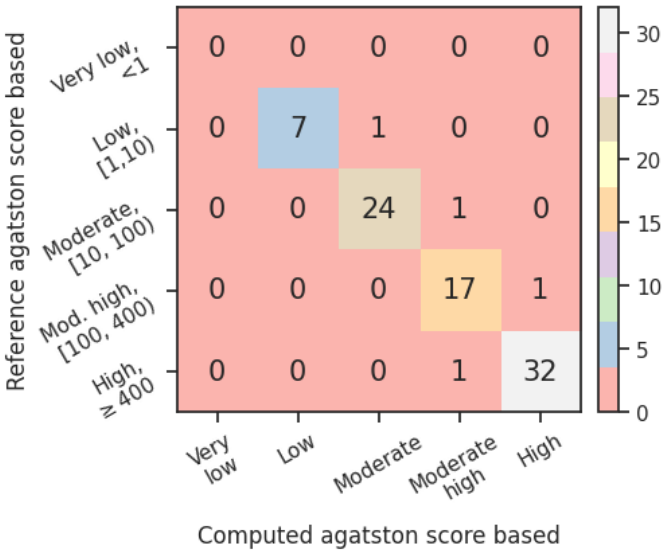

FIGURE 18 – Confusion matrix between the different risk level for the 2D version

As we can see from this confusion matrix, the model shows a slight tendency to overestimate the risks. Specifically, there are three instances where a '1' appears on the upper diagonal, meaning the model predicted a higher risk class than the actual one. From a medical standpoint, this is debatable, as the goal is to achieve maximum precision. However, if we must choose between overestimating and underestimating, I believe overestimating is the lesser issue.

The results obtained with this 2D model have far exceeded our expectations and are very promising. The Dice coefficient and accuracy score are significantly higher than those of previous methods, leading to improved reliability in this step, which in turn enhances the overall process.

Metric	Value
CSV Filename	Result_Dataset004_COCA2Dv3.csv
Model Name	COCA2Dv3
Pearson Correlation (r)	0.9221
Pearson Correlation p-value	1.45 × 10^-35
Pearson Correlation Confidence Interval	[0.8821, 0.9489]
Bootstrapped Pearson Correlation (Lower)	0.8717
Bootstrapped Pearson Correlation (Upper)	0.9726
Spearman Correlation (r)	0.9948
Bootstrapped Spearman Correlation (Lower)	0.9895
Bootstrapped Spearman Correlation (Upper)	1
Regression Line Slope (m)	1.0501
Regression Line Intercept (b)	12.7615
Weighted Kappa	0.9573
Kappa Confidence Interval	[0.9381, 0.9764]
Max Value in Dataset	6236.45

TABLE 1 – Metrics for COCA2Dv3 Model

7.2 3D Version

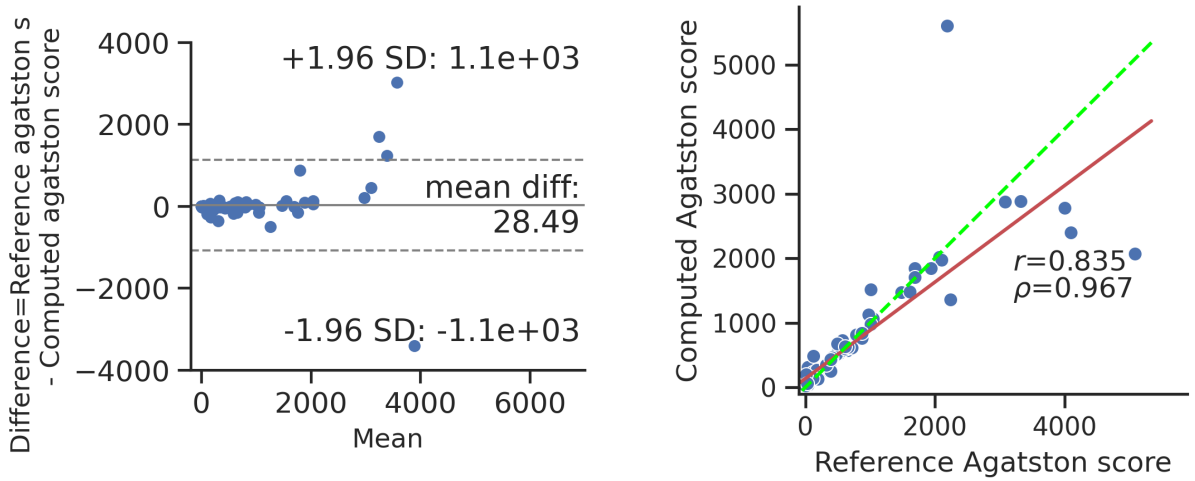

FIGURE 19 – Comparison of the given Agatston score and the derived Agatston score for the 3D version

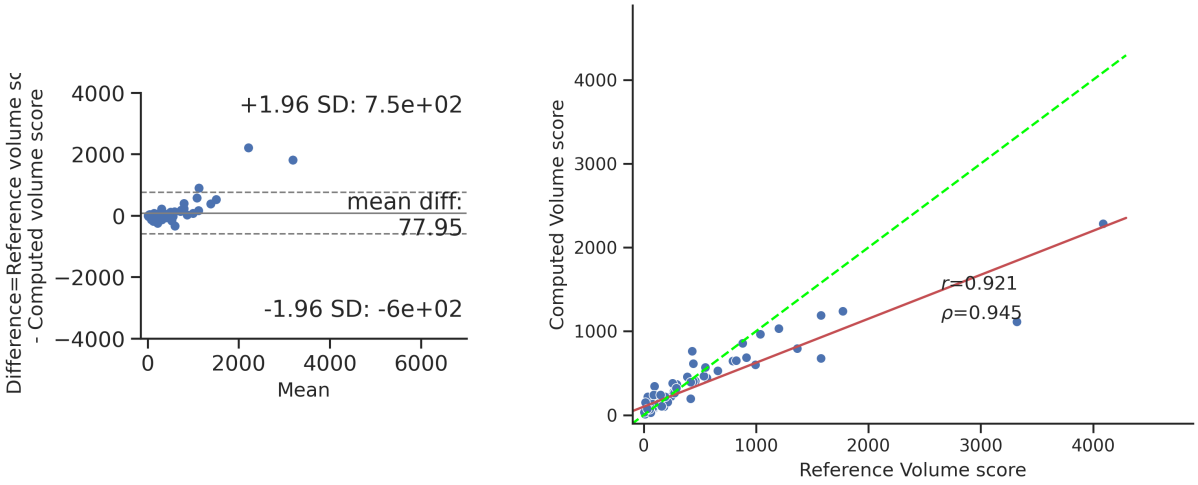

FIGURE 20 – Comparison of the given Volume score and the derived Volume score for the 3D version

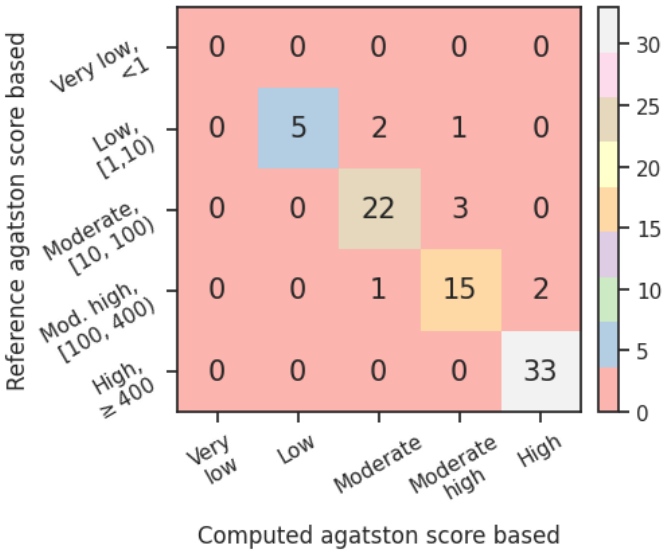

FIGURE 21 – Confusion matrix between the different risk level for the 3D version

A travers cette matrice de confusion on peut voire que comme pour la methode 2D le model a tendeance a surestimer plus que sous estimer les risques.

We obtained very conclusive results that also outperform the performance achieved with previous methods.

Metrics	Values
CSV Filename	Result_Dataset002_COCA3D.csv
Model Name	COCA3D
Pearson Correlation (r)	0.835
Pearson Correlation p-value	5.52 × 10^-23
Pearson Correlation Confidence Interval	[0.7560, 0.8901]
Bootstrapped Pearson Correlation (Lower)	0.7648
Bootstrapped Pearson Correlation (Upper)	0.9053
Spearman Correlation (r)	0.9666
Bootstrapped Spearman Correlation (Lower)	0.9496
Bootstrapped Spearman Correlation (Upper)	0.9837
Regression Line Slope (m)	0.7473
Regression Line Intercept (b)	136.231
Weighted Kappa	0.8913
Kappa Confidence Interval	[0.8535, 0.9290]
Max Value in Dataset	5598.73

Table 2 – Metrics for COCA3D Model

8 Comparison

Metrics	COCA2Dv3 (95% CI)	COCA3D (95% CI)
Pearson's r	0.9221 (0.8821 - 0.9489)	0.8350 (0.7560 - 0.8901)
Spearman's	0.9948 (0.9895 - 1.0000)	0.9666 (0.9496 - 0.9837)
Cohen's (Weighted Kappa)	0.9573 (0.9381 - 0.9764)	0.8913 (0.8535 - 0.9290)

Table 3 – Comparison of COCA2Dv3 and COCA3D models based on different metrics

This table shows that across different performance metrics, the best results were achieved using the 2D method. This is somewhat counterintuitive, as transitioning to 2D inevitably results in the loss of spatial information. This could suggest that such information may not be crucial for calcium detection, or that the 3D model is too complex to effectively process all the available data. The 2D model is much lighter and faster in training while still delivering better performance.

9 Conclusion

Contrary to our initial assumptions, the 2D version of the model provided the best performance in terms of both accuracy and computational efficiency. Despite the 3D version's ability to retain more spatial information, the 2D approach was more effective for this task, achieving higher correlation scores with a lower computational cost. The results highlight that in certain cases, simpler models can outperform more complex ones, especially when the additional information does not significantly enhance the model's predictive capability.

The use of the COCA dataset was essential for training and validating the models, and the strong performance metrics achieved suggest that the 2D approach could be a valuable tool for automated coronary calcium scoring. Moving forward, efforts could focus on refining this 2D model and testing its generalizability across other datasets. Unfortunately, due to delays in obtaining access, I was unable to test my model on the Rotterdam Study database from Erasmus MC.

Finally, I thoroughly enjoyed working on this project, which allowed me to gain a deep understanding of various concepts in the field of machine learning. Additionally, I was very pleased with both the work environment and the city of Rotterdam itself. The supportive team and vibrant atmosphere greatly enriched my experience. I hope that the results I achieved can contribute to future projects and serve as a foundation for further advancements in this area.

L'enssemble de mon projet etant diponible sur Gitub : `https://github.com/Diabostyle`

10 References

nnUnet -`https://github.com/MIC-DKFZ/nnUNet`

Stanford **COCA Database** - `https://aimi.stanford.edu/datasets/coca-coronary-calcium-d`

Barda, N., Dagan, N., Stemmer, A., Yuval, J., Bachmat, E., Elnekave, E. (2022). **Prediction of Coronary Artery Disease via Calcium Scoring of Chest CTs.** `https://cardiooncologyjournal.biomedcentral.com/articles/10.1186/s40959-023-00196-9`

Eng, D., Chute, C., Khandwala, N., Rajpurkar, P., Long, J., Shleifer, S., Khalaf, M. H., et al. (2021). **Automated coronary calcium scoring using deep learning with multicenter external validation.** `https://www.nature.com/articles/s41746-021-00460-1`

Lessmann, N., van Ginneken, B., Zreik, M., de Jong, P. A., de Vos, B. D., Viergever, M. A., Isgum, I. (2018). **Automatic calcium scoring in low-dose chest CT using deep neural networks with dilated convolutions.** IEEE Transactions on Medical Imaging, `https://ieeexplore.ieee.org/document/8094970`

Pölsterl, S., Wolf, T. N., Wachinger, C. (2021). **Combining 3D Image and Tabular Data via the Dynamic Affine Feature Map Transfer.** Proceedings of MICCAI 2021. `https://link.springer.com/book/10.1007/978-3-030-59719-1`

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). **Grad-CAM : Visual Explanations from Deep Networks via Gradient-Based Localization** https://ar5iv.labs.arxiv.org/html/1610.02391

Zeiler, M. D., Fergus, R. (2014).**Visualizing and Understanding Convolutional Networks. European Conference on Computer Vision (ECCV)** https://ar5iv.labs.arxiv.org/html/1311.2901