Prof. Pietro Cerveri

Tutor: Matteo Leccardi
matteo.leccardi@polimi.it

NEUROENGINEERING 2023-2024 – PROJECT WORKSHOP #3

MAPPING VESSELS AND HEART CORONARY ARTERIES FROM CCTA IMAGES FOR CORONARY HEART DISEASE DIAGNOSIS

Clinical Context

Heart disease and stroke claim more lives each year than all forms of cancer and chronic lower respiratory disease combined. The most frequently performed invasive diagnostic procedure in patients with suspected coronary artery disease (CAD) is invasive coronary angiography (ICA), which is shown to be income dependent. On the other hand, coronary computed tomography angiography (CCTA) provides a non-invasive, fast, and relatively cheap means of examining the coronary arteries. Due to the high clinical significance, sensitivity, and ease of acquisition of contrast liquid-enhanced CT images compared to invasive procedures, CCTA are being investigated to assess the significance of atherosclerotic plaques on blood flow dynamics with very promising results, de facto paving the road towards democratizing patient-tailored CAD assessment.

Technical Context

The first step in each and every CCTA-based CHD assessment procedure is to map the arterial tree. This can be achieved either by segmenting the coronary arteries, or by finding the geometric center of the vessels, called a "center line" or "centerline". We will focus on the extraction of the centerlines from CCTA images since it allows us to obtain a full map of the arterial tree as well as making subsequent segmentation easier and more precise. Moreover, arterial centerlines can be used as a standalone tool to visualize planar reformatted volumes which rectify the artery and allows a first quick evaluation of the patient by a trained expert clinician.
This technical challenge can be subdivided into several sub-challenges, such as:

- identification of whether a point, placed into the image space, lies inside a coronary artery,
- identification of whether a point, placed inside or near enough a coronary artery, lies:
    - at the coronary ostium, which is the interface between the aorta and the coronary artery,
    - at the intersection between arterial branches,
    - at a generic point in the artery which is not close to either the coronary ostium or an intersection
- lumen radius regression given a point inside the artery,
- identification of the direction (unit vector in R^3) a point should be displaced towards to get closer to the centerline, given a point inside of or near enough to the coronary artery; the direction of the unit vector should be perpendicular to the coronary artery,
- classification of the arterial trees (left or right) by their characteristic shapes, given a reconstructed arterial tree,
- semantic segmentation of the reconstructed arterial tree to assign each arterial segment a name, such as Right Coronary Artery, Left Coronary Artery, Left Anterior Descending, Left Circumflex, etc.

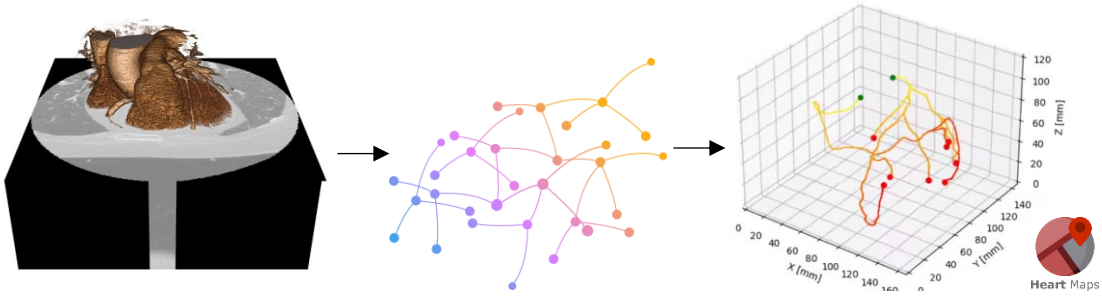
AIM of the work

Recent advances in Deep Learning can be leveraged to achieve all this. This work aims to develop a Convolutional Neural Network (CNN) fully automatic framework working on CCTA images that will be able to map the coronary artery tree. Specifically, each group will tackle one of the following sub-problems:

- artery radius regression,
- point belonging to coronary artery binary classification,

- point in artery belonging to coronary ostium, arterial intersection, or arterial segment multiclass classification,
- prediction of the unit vector pointing to the nearest centerline posed as a multiclass classification problem.

For the most ambitious, the following challenge is proposed: arterial tree classification (left/right) by using a graph neural network (it would be advisable to implement the attention mechanism so that the network can consider the whole tree before taking decisions). This task is at the current state-of-the-art level in this research field and would represent a proper challenge even for your tutor, so choose it only if you have enough time to put into it, considering it would require a deeper state-of-the-art analysis and advanced coding skills to complete it in time.



Datasets

CCTA images from publicly available datasets (CAT08 and ASOCA challenges from MICCAI 2008 and 2020, https://grand-challenge.org/) will be provided to achieve the proposed tasks, along with the centerline map represented as a graph structure and some utilities to easily interface with the datasets.
Additionally, synthetic data may be provided should a group choose to opt for a transfer learning paradigm.

References

Reference papers will be made available on the WeBeep platform inside the folder of the project.
As for the more challenging arterial tree classification task, a starting set of reference papers will be made available to the group, if any, after the choice of the challenge. It has to be considered that this project will require autonomous research of the state-of-the-art methods, and requires overall a significative higher level of autonomy with respect with the other project proposals.

Support material

Model development will require the use of Nvidia GPUs. If you have one on your desktop, feel free to use it. An option is available (and preferred in most cases) to use Google Colab (https://colab.research.google.com/). Google Colab, as the name implies, allows for collaborative coding.
Networks will be developed in pytorch (https://pytorch.org/tutorials/), which can be installed also in Google Colab (https://pytorch.org/tutorials/beginner/colab.html , https://medium.com/analytics-vidhya/google-colab-tutorial-how-to-setup-a-deep-learning-environment-on-colab-bc5ab7569f02). Pytorch is a valuable tool not only for training and working with neural networks of any sort, but also for physics simulations and high performance computing thanks to its direct integration with Nvidia GPUs and its automatic differentiation engines.

A library to work with graph structures will be provided (https://github.com/AAMIASoftwares-research/HCATNetwork/tree/main based on https://networkx.org/documentation/stable/index.html#), as well as a library to work with the datasets seamlessly (https://github.com/AAMIASoftwares-research/DatasetUtilities).

As students, you might be entitled to use Github Copilot for free. Github Copilot is an AI which helps you with code (https://github.com/features/copilot). Copilot might be a valuable tool in completing this project as it speeds up slow coding tasks and lets you focus on the bigger picture. To activate it, make a github account, activate Github student (https://education.github.com/discount_requests/application), follow the instructions (as a certification method, you can enter the online services, take a screenshot at the whole page making sure that your name and matriculation

number is visible, and use that screenshot), wait a couple of days for activation of student benefits, and then activate Copilot. Once3 active, in VS Code or other software, activate the Copilot extension, and start using Copilot. Open an empty file in VS Code and try following Copilot's instructions.

Note that Copilot should not be available on Google Colab, however it works natively with Visual Studio Code (available for all OS, even Linux). Copilot works better in Microsoft's VS Code, as some features are not yet available on other editors.