POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Deep Learning based Coronary Artery Centerline tracking aimed at Fractional Flow Reserve Prediction from CCTA images

LAUREA MAGISTRALE IN AUTOMATION AND CONTROL ENGINEERING - INGEGNERIA DEL CONTROLLO E DELL'AUTOMAZIONE

Author: MATTEO LECCARDI

Advisor: PROF. MARCO MARCON

Co-advisor: PROF. GIAN FRANCO GENSINI, PH.D. MARCO BRANDO MARIO PARACCHINI

Academic year: 2021-22

1. Introduction

Coronary artery disease (CAD) is characterized by an inadequate supply of oxygen-rich blood to the myocardium caused by the narrowing or blocking of a coronary artery due to the build-up of cholesterol and fatty deposits on the inner lining of the arterial wall. In extreme cases, the effect may be a myocardial infarction. CAD accounted for approximately 12.6% of deaths in the US in 2018, causing 360'900 deaths and being the leading cause of death on the world [3].

Fractional Flow Reserve (FFR) assessment is a relatively novel invasive procedure for CAD evaluation based on blood pressure measures inside the coronary vessel, which proved to be more effective than other standard invasive procedures such as Coronary Angiography. Lately, many approaches are being proposed to fully-automatically estimate the FFR index directly from Coronary Computed Tomography Angiography (CCTA) images (FFR_{CT}). The severity and anatomical extent of CAD can be then assessed non-invasively by CCTA, which can provide both qualitative information and quantitative information such as calcium scoring and FFR_{CT} . Current state-of-the-art methods rely on computational fluid dynamic to simulate blood flow in the coronary arterial tree of interest and derive the FFR_{CT} virtual measure, however lately some new approaches based completely on deep leaning image analysis techniques showed very promising results [5].

Whichever the case, all methods rely on preliminary automated coronary artery centerlines estimation. Recently, Wolternik et al. [4] proposed a convolutional neural network (CNN) based tracking scheme which iteratively tracks the location of the centerline by following the vessel starting from a manually- or automatically-placed starting point (seed) in the vessel of interest. State-of-the-art performance was achieved with respect to any centerline extraction algorithm based on deep learning techniques. Their coronary artery tracker required multiple seeds per arterial branch to successfully extract the related centerlines, and fails if too few seeds points are initialised - in the limit case, one point per arterial branch.

Building on [4], this work proposes three new CNN architectures, as well as a new tracker capable of reconstructing the full arterial branch (left or right) starting from just one seed point planted inside a coronary artery. It is shown that both the proposed CNN and the tracker obtain better performance scores in tests performed locally on the openly available CAT08 dataset [2]. The advantages of adopting a highly modular tracker through object-oriented programming are discussed. Moreover a post-processing step based on graph-algorithms is introduced as a possible solution to automatically classify and reconstruct a topologically consistent map of the left and right coronary artery trees starting from the cloud of points resulting from the tracking procedure. Two standard test suites are built to test and compare the proposed CNNs and trackers: one concerning the quantitative evaluation of the proposed CNNs and competition against Wolternik's, the other concerning the qualitative evaluation of the tracking capability of the trackers in difficult conditions and rough quantitative estimation of tracking capability and accuracy, inspired by [2].

2. Methods

The proposed method can be subdivided in three contributions: three novel CNN structures, a new, modular centerline tracker, and a graph-based post-processing framework.

2.1. New CNN architectures

The novel CNN structures are inspired by the model proposed by Wolternik et al [4], who proposed to determine the orientation and radius of a coronary artery at a location x in a CCTA image I based on a 3D isotropic image patch P with side 9.5 mm. The estimated direction and radious are then used by a tracker to compute the displacement vector towards the next location to follow in the iterative tracking process. The network contains in total 7 layers: 5 convolution layers of which two have dilation 2 and 4, and two fully connected layers at the output side. The output layer of the CNN consists of classification nodes that determine a posterior probability distribution over possible tracking directions D and a regression node that determines the radius R of the vessel at point x. The same convolutional feature extraction filters are used to both classify directions and estimate the radious, which makes the network compact but requires that the shared weight are adjusted to perform both tasks. Sharing weights to this extent might lead to an undersized network which has to compromise too much to perform both tasks well enough.

Input-layer-split. The first investigated network is a composition of two of Wolternik's networks, each of which is specialised in the execution of either the classification or regression tasks. This results in a network double the size of Wolternik's [4], which however gives more freedom to the model parameters to learn the specific tasks independently. Results show that this path is viable and indeed improves the network performance, and should be further investigated to reduce its dimension and further increase performance.

Midsplit-4. The second investigated network is built on the basis of Wolternik's [4] up until the fourth layer, after which the network is split in two equal and independent branches, one concerning the direction classification task, the other the radious regression. The main purpose of splitting the network after the first 4 layers is to let the network learn the basic geometric features useful for both tasks, and let the high-level filters adjust to interpret such features to perform one precise task, independently. This approach is common in the visual cortex of mammals and in convolutional neural network for image processing, where the basic features such as edges and boundaries are expected to be the same in any context, while the high-level features extracted from the low level ones are expected to change with respect to the task.

Midsplit-4-FD. Since it was found that the quality of the centerline tracking task greatly depends on the quality of the radious regressor, the possibility of adding knowledge regarding the direction predicted by one branch of the network into the radoius regression branch is explored in the third proposed network architecture. The rationale behind this choice is the assumption that, once the most likely directions are known, the displacement vector's modulus (the estimated radious R) can be estimated more precisely by using the knowledge about the direction distribution fed back to the radious regression branch. To better process the newly added information, another fully connected layer is added to the radous regression branch (figure 1).

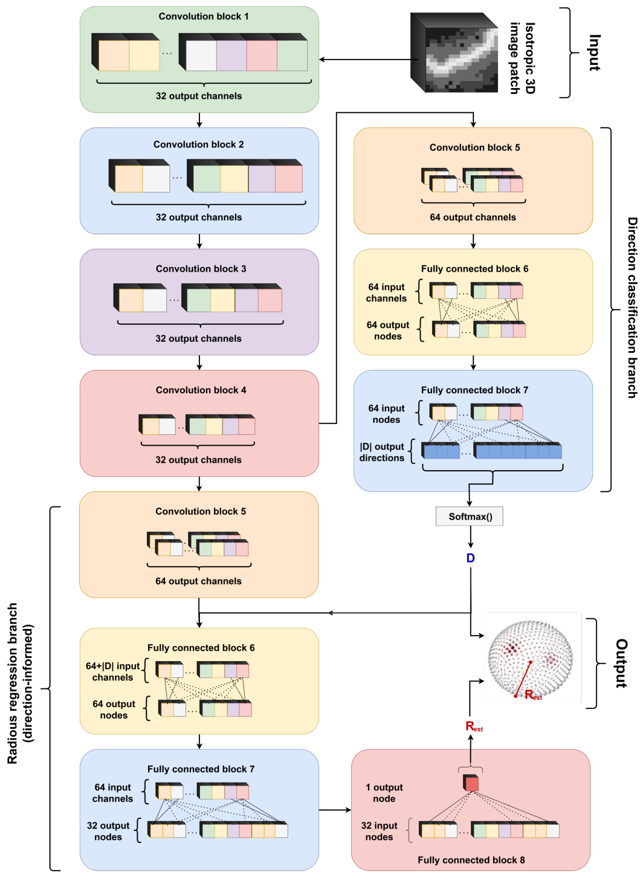

Figure 1: Structure of the midsplit-4-FD network.

2.2. Dataset and training strategy

To train, test, and compare the proposed networks and tracker, the openly available Rotterdam Standard Evaluation Framework dataset [2] is used. In each scan, the centerline and radius of four major coronary arteries were manually annotated in a consensus reading by three experts. The standard training set consists of 8 CCTA images with reference annotations for the centerline location and radius, and the test set consists of 24 CCTA images for which no reference annotations were openly distributed. The standard test procedure would require to locally extract centerlines from the 24 test images and then submit the results to a standardised online evaluation tool [2]. Due to the current unavailability of the standard online evaluation tool, the network was trained on 7 of the 8 annotated CCTA scans, and the left out image was used for validation.
The labelled dataset for training and testing is

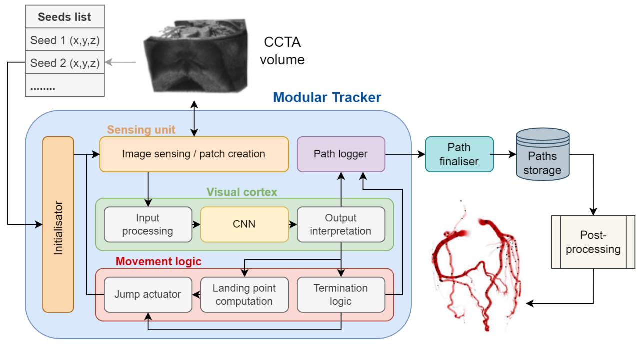

Figure 2: Representation of the tracker and its modules (boxes) and interfaces (arrows). Any number of additional modules could be added to the tracker and connected to any other component through interfaces.

created as explained in [4], with some minor changes. Training is performed in the same conditions for all tested networks, so that they can be fairly compared.

2.3. Trackers

The tracking algorithm is at the core of the automatic extraction of the centerlines. Through object oriented programming, it is convenient to define a tracker class which takes care of all the aspects of the tracking procedure and inside of which any of the proposed networks could be plugged in at will through an interface. Any other tracker can be created as an extension of the basic tracker architecture, shown in figure 2.
The tracker needs a neural network to interpret its surrounding and make predictions on what are the viable paths, however the network needs the tracker to "move along" the centerline and hop from the current center of the patch x to the next one through the displacement vector.
The tracking algorithm starts at a seed point x_{seed}. An isotropic 3D patch P_0 centered at x_{seed} is extracted by the tracker and processed by the CNN. The posterior probability distribution $p(D|P_0)$ over all possible directions and the estimated radius R_0 are obtained from the model in use (in this case, the CNN).
To determine two initial opposing directions of the tracker for its first jump, two local maxima d_0 and d_0' separated by an angle $\geq 90°$ are identified in $p(D|P_0)$. The tracker will first follow the centerline in the d_0 direction until termination, and then in the direction d_0' until termination.
To follow the centerline in the direction d_0, the

tracker takes a step of specified length towards d_0 and arrives at point x_1. The step size can depend on many factors and any tracker can define its logic to determine the step size, however the considered tracker takes R_0 as the step size. The tracker makes sure to land in a pixel different from the current center of the patch P by increasing the step size, in necessary.
Then, a new patch P_1 is extracted at x_1 and processed by the CNN to provide $p(D|P_1)$ and R_1. The new tracking direction d_1 is selected as the direction with the highest probability in $p(D|P_1)$. To prevent the tracker from moving backwards, directions that have an angle $\geq 90°$ to d_0 are excluded by default from the selection. In the tracker implemented by Wolternik et al. [4] the value is lowered from 90° to 60°. This process is repeated until a stopping criterion is fulfilled. Subsequently, the tracker follows the same process in the second initial direction d_0', starting again at the seed point x_{seed}.
Termination of the tracker is guided by a stopping criterion. Each tracker can implement its own stopping criterion thanks to the intrinsic modular structure, which can be based on hard constraints or learnt conditions as well. The trackers considered in this work all implement a stopping criterion based on the uncertainty of the direction classifier, as proposed by Wolternik et al. [4]. At each point along the extracted centerline, the normalized entropy $H(p(D|P)) \in [0,1]$ of the posterior probability distribution is computed:

$$\theta_H = H(p(D|P)) = \frac{\sum_{d\in D} -p(d|P)\log_2 p(d|P)}{\log_2 |D|}$$

The movement terminates if the trailing average over the last three steps of the entropy of the selected probability distribution crosses a threshold value $\theta_H = 0.9$, as implemented by Wolternik et al. [4]. The tracker interfaces with a class that saves the path, radii, and termination criterion value for each branch of the movement into a data structure for each seed.
Wolternik's tracker performs well when enough seeds are planted in the portions of interest of the coronary artery tree, however it performs quite poorly when starting from few seeds, the limit case being when it starts from just 2 seeds.
Seedspawner tracker. A possible enhancement to the tracker proposed by Wolternik et al. [4] is to add some new functionality that make it more effective in case a low number of seeds (1 per coronary artery branch) are initiated.
A simple solution would be to add new seeds iteratively to the original list of seeds through a "seeds spawner" module, which then grows exponentially and allows for a more comprehensive exploration of the arterial tree. The seeds spawning is composed of two modules, each adding new seeds to a queue.
The first module spawn seeds in a Hansel and Gretel fashion: at each tracker hop from x_n to x_{n+1}, 5 new seeds are spawned in the surroundings of the tracker and added to the queue with exponentially decreasing probability starting at 0.1. A uniform distribution is used instead of gaussian to scatter the seeds because the main interest of this module is to place seeds far enough to have a chance of falling into unexplored side-branches, for example when bifurcations occur. The second module tries to place new seeds in the direction of the most likely movement directions in a probabilistic way. The idea is to detect all the most likely 25% direction predicted by the CNN and "throw" a seed in each of such directions when the termination entropy is sensed to be between 0.85 and 0.93, which should happen near bifurcations and obstacles. Any seed is placed in the queue if and only if no seed in the original list or in the queue is less than 4 mm away from the new seed. The seeds are actually added to the original seeds list only if the length of the final tracked path is more than 5 jumps long: this condition helps not placing seeds where the tracker is uncertain of the direction to follow because the termination criterion is met almost right away in the tracking procedure. A cap of 200 total seeds is imposed, after which no new seed can be added to the list of seeds from which to start exploring.

2.4. Post processing

What is obtained from the centerline tracking procedure is an unstructured set of filaments which retain no information on their membership to the left or right coronary arterial branch. The goal of the proposed post processing procedure is to classify each point as part of either the left or right arterial tree, and for each tree to build an oriented graph that, starting from the coronary ostium, connects the whole arterial tree in a single, neatly organised data structure.

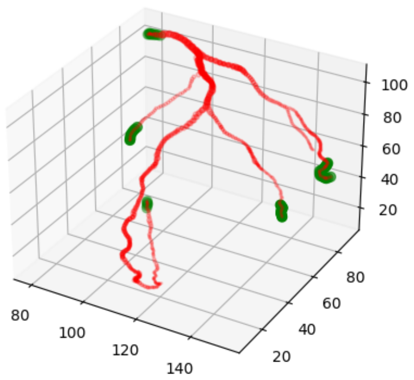

Figure 3: Green endpoints automatically classified in an automatically subdivided arterial tree (red).

Classification of each point in one of the two arterial trees (left or right) can be performed using the well-known DBSCAN algorithm. Once every point is classified, the procedure works on one arterial tree at a time. All endpoints of the filamentous structure are obtained by considering for each point the distribution of the versors pointing towards its 20 nearest neighbours which are at least 4 mm away from the considered point. Versors are considered as 3D points and again clustered with the DBSCAN algorithm. Versors of endpoints obtained from real endpoints would be clustered in just one class, while all the other points would be clustered into 2 or more classes. The results of the automatic arterial branch subdivision and the endpoints classification is shown in figure 3.
It is then enough to understand which of the endpoints have the nearest neighbours with highest maximum estimated radious to find the coronary ostium, while all the others are considered endpoints. Then, all the points are transformed into a graph and connected to their 6 nearest neighbours. The connection weight is defined as the euclidean distance between the two connected nodes. The shortest path between the ostium point and each one of the endpoints is found through the A* algorithm. The shortest patch passes mostly to the center of the filament point cloud, which should enhance the quality of the centerline estimation. However when hard turns are presented it drifts a little towards the minimum-length side of the curve. Subsequent extracted paths can be combined easily together by merging the nodes in common between two paths up until an eventual bifurcation happens, at which point one node will be connected to two other nodes.
The proposed post processing procedure is still a prototype, however it has the potential to not only offer a ordered representation of the centerline tree, but also to improve centerline extraction accuracy by making the final extracted path pass through the center of the filamentous cloud of points surrounding the real centerline.

3. Tests and results

The creation of two evaluation frameworks were necessary for two reasons: the main reason is that, during fast prototyping of the networks and trackers, a quick way to asses their quality is needed. The second reason is that the Rotterdam standard evaluation framework, which should be reachable from the Biomedical Imaging Group Rotterdam official website `https://bigr.nl/`, seems to be offline at present day.
Two evaluation frameworks are presented. The *local test suite* is used for assessing the quality of the trained neural networks in a quantitative way. When testing networks architectures against each other, the networks were trained all in the same conditions. First, a standard set of test datapoints is generated around the centerlines in a procedural way, where each test point is unique in the set and corresponds to the center of a unique image patch to be tested; for each datapoints some useful information are stored such as the proximity to the annotated centerline, reference radious used in training for that specific datapoint as well as the reference directions. The second stage of the test is to let the CNNs perform predictions about the directions and radious by using as inputs the patches centered on the set of test points. The last stage of the local test performs a fully automated quantitative analysis of the extracted data to compute, store and visualise meaningful metrics which aim at assessing the radious estimation error, the direction estimation (angular) error, the distance from each estimated landing point $P_{landing}$ and its nearest centerline point $P_{centerline}$, and the distance from each estimated landing point $P_{landing}$ and its reference point of the centerline on which the network was trained on, $P_{centerline,opt}$ [4]. The last two metrics are the most important ones as they allow to assess

the potential tracking accuracy of a CNN and how well the CNN can generalise training, respectively.
The *global test suite* on the other hand is used for assessing the quality of the centerline tracking and reconstruction mostly, but not entirely, in a qualitative way. Qualitative inspection of the extracted centerlines is performed, as well as a rough estimation of the metrics used in the Rotterdam Evaluation Framework [2] "OV", "OF", "OT", "AI" which measure the tracking capability (OV, OF, OT) and the tacking accuracy (AI) of the algorithm.

3.1. Results

Evaluation of the proposed networks. The local tests performed on the faithful re-implementation of Wolternik's network [4] showed a mean radious estimation absolute error mean($R_{err,abs}$) of 0.246 mm, a mean direction estimation error mean(α_{err}) of 25.7°, and most important of all a mean distance between the network's predicted landing point $P_{landing}$ and the closest centerline point $P_{centerline}$ (mean($d(P_l, P_c)$)) of 0.402 mm.
On the other hand, local tests performed on the *input-layer-split* network trained and tested in the same exact conditions of Wolternik's showed a mean($R_{err,abs}$) of 0.197 mm, a mean direction estimation error of 25.18°, and a mean distance between $P_{landing}$ and $P_{centerline}$ of 0.38 mm, which means an increase in performances of circa 19.9%, 2% and 5%, respectively.
Local tests performed on the *midsplit-4* network trained in the same conditions of Wolternik's [4] and input-layer-split showed a mean($R_{err,abs}$) of 0.158 mm, a mean direction estimation error of 24.0°, a mean distance between $P_{landing}$ and $P_{centerline}$ of 0.379 mm, and an overall better consistency of the predictions which means performance is once more increased with respect to both Wolternik's [4] and the input-layer-split network. This shows that Wolternik's network [4] was indeed undersized and that sharing just the first 4 layers not only reduces the size of the network by 33% with respect to the input-layer-split network, but it also increases its performance. Most noticeably, this network is capable of better predicting the correct $P_{landing}$ with respect with the optimal landing points it was trained on, showing a performance increase of

CNN name	Domain	$\bar{R}_{err,abs}$ [mm]	$\bar{\alpha}_{err}$ [deg]	$\bar{d}(P_l, P_c)$ [mm]	$\bar{d}(P_l, P_{c,opt})$ [mm]	Learnable params	Test exec. time [s]
Wolternik's	G	0.246	25.74	0.402	0.515	**176'533**	**110**
	AO	0.203	24.81	0.368	0.466		
Input-layer-split	G	0.197	25.18	0.380	0.472	353'066	168
	AO	0.152	24.56	0.346	0.422		
Midsplit-4	G	0.158	**24.04**	0.379	0.454	236'309	123
	AO	0.114	**23.70**	0.347	0.419		
Midsplit-4-FD	G	**0.1399**	25.26	**0.3498**	**0.424**	270'357	133
	AO	**0.097**	24.58	**0.313**	**0.374**		

Table 1: Table showing the most relevant metrics for evaluating the tested CNNs. the proposed Midsplit-4-FD network outperforms Wolternik's network [4] in all the relevant metrics. values indicated with $\bar{value}$ are averages over the considered set of test points. Here two sets of test points are considered to compute the petrics: "G" standing for all the test points, and "AO" standing for "Away from Ostia", which are the points being at least $1.1R$ mm away from the coronary ostium.

11.8% and 3.6% with respect to Wolternik's [4] and input-layer-split networks, respectively.
Local tests performed on the *midsplit-4-FD* network trained in the same conditions of the previously discussed networks showed a mean($R_{err,abs}$) of just 0.14 mm, a mean direction estimation error of 25.26°, a mean distance between $P_{landing}$ and $P_{centerline}$ of 0.35 mm, which means another increase of general performance with respect to the previous networks. This network can predicting the correct $P_{landing}$ with respect with the optimal landing points the network was trained on better than the midsplit-4 network, showing a performance increase of 17.6% and 6.6% with respect to Wolternik's [4] and midplit-4 networks, respectively.
Table 1 proposes a general overview of the 4 discussed networks with their characteristics and benchmark test results.
Evaluation of the proposed trackers. Three trackers competed against each other in the global test suite to extract centerlines starting from 5 seeds and from 2 seeds: a basic tracker (which is not relevant in this discussion), Wolternik's tracker [4] and the *seedspawner tracker*. All trackers were equipped with the same neural network (Wolternik's [4]) for fair competition and started tracking the centerlines from the same exact 5 and 2 seeds.
Wolternik's tracker [4] performed overall quite well in the 5 seeds challenge, however it performed insufficiently in the 2 seeds challenge (figure 4). The two arteries branching from the

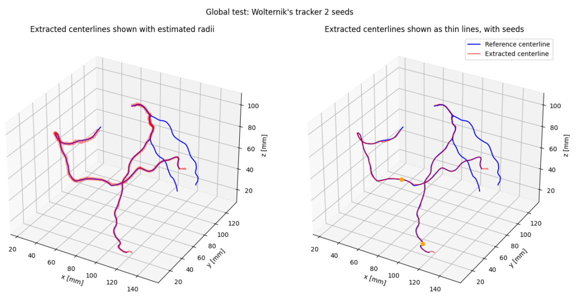

Figure 4: Centerlines extracted from 2 seeds with Wolternik's tracker [4]. Two of the four reference vessels are completely ignored. Blue: reference. Red: extracted.

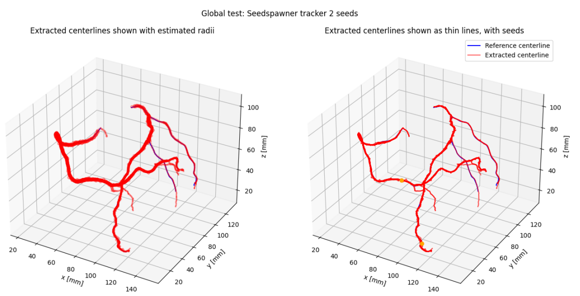

Figure 5: Centerlines extracted from 2 seeds with the seedspawner tracker. Blue: reference. Red: extracted.

	Wolternik's, 2				**Seedspawner, 2**			
	v1	v2	v3	v4	v1	v2	v3	v4
OV %	99.52	100	21.11	56.21	100	100	98.6	100
OF %	0 *	100	21.11	56.21	100	100	98.6	100
OF [mm]	0 *	176	22	61	224	176	104	109
OT %	99.45	100	42	97.8	100	100	100	100
AI [mm]	0.657	0.51	0.698	0.670	0.301	0.267	0.365	0.342

Table 2: Table reporting the computed metrics for each reported test, for each vessel. The blue metrics give an idea of the tracking ability, while the orange metric of the tracking accuracy [2].

left main artery are not explored. On the other hand, the proposed seedspawner tracker managed to reconstruct the whule arterial tree starting from the same 2 seeds (figure 5). It also managed to find non-annotated vessels. Table 2 proposes a semi-quantitative overview of the tests performed on the trackers.[1]

[1]Values of 0 in OF are encountered when, right from the start (coronary ostium) the tracked centerline is out of the reference centerline radious. It means that the tracker deviates from the correct course when it gets near the coronary ostium. This phenomenon was not observed in the Seedspawner tracker.

4. Conclusions

An automatic deep learning based coronary artery centerline tracker in CCTA images was proposed, as well as a three new CNN architectures to drive the proposed tracker. The inspiration leading to the adopted design choices of the proposed CNNs is found in the structure of the visual cortex of mammals and in many network's architecture found in recent literature, where the first few layers are used to extract low-level information such as edges or corners and curves, while the last layers are used to derive problem specific knowledge from information passed from the previous layers. The local and global tests performed in a standardised way for both the baseline CNN and tracker proposed by Wolternik [4] re-implemented locally, and for the proposed CNNs (midsplit-4-FD in particular) and tracker (seedspawner) showed that both the proposed CNN and the proposed tracker score higher performance indexes with respect to the re-implemented baseline.
Overall, the proposed tracking scheme improves the one proposed by Wolternik et al. in 2018 [4] by proposing a better performing network tested locally and implement a very simple extension to the tracker basic tracker logic for discovering new viable paths by planting seeds along the way while tracking the vessel's centerline, as opposed to the more elegant yet complicated bifurcation classification network proposed by Salahuddin et al. in 2021 [1], which needs a new set of annotated data to learn to detect bifurcations.

In the near future, hopefully the proposed trackers will be the basis for the coronary artery extraction system used in the fully automatic estimation of the Fractional Flow Reserve index non-invasively from CCTA scans. Hopefully, with enough training data, the new FFR_{CT} estimation approaches can be trained on CCTA which do not even require the hyperaemia condition.

References

[1] Zohaib Salahuddin, Matthias Lenga, and Hannes Nickisch. Multi-resolution 3d convolutional neural networks for automatic coronary centerline extraction in cardiac ct angiography scans. In *2021 IEEE 18th Inter-*

national Symposium on Biomedical Imaging (ISBI)*, pages 91–95, 2021.

[2] Michiel Schaap, Coert T. Metz, Theo van Walsum, Alina G. van der Giessen, Annick C. Weustink, Nico R. Mollet, Christian Bauer, Hrvoje Bogunović, Carlos Castro, Xiang Deng, Engin Dikici, Thomas O'Donnell, Michel Frenay, Ola Friman, Marcela Hernández Hoyos, Pieter H. Kitslaar, Karl Krissian, Caroline Kühnel, Miguel A. Luengo-Oroz, Maciej Orkisz, Örjan Smedby, Martin Styner, Andrzej Szymczak, Hüseyin Tek, Chunliang Wang, Simon K. Warfield, Sebastian Zambal, Yong Zhang, Gabriel P. Krestin, and Wiro J. Niessen. Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. *Medical Image Analysis*, 13(5):701–714, 2009. Includes Special Section on the 12th International Conference on Medical Imaging and Computer Assisted Intervention.

[3] Connie W. Tsao, Aaron W. Aday, Zaid I. Almarzooq, Alvaro Alonso, Andrea Z. Beaton, Marcio S. Bittencourt, Amelia K. Boehme, Alfred E. Buxton, April P. Carson, Yvonne Commodore-Mensah, Mitchell S.V. Elkind, Kelly R. Evenson, Chete Eze-Nliam, Jane F. Ferguson, Giuliano Generoso, Jennifer E. Ho, Rizwan Kalani, Sadiya S. Khan, Brett M. Kissela, Kristen L. Knutson, Deborah A. Levine, Tené T. Lewis, Junxiu Liu, Matthew Shane Loop, Jun Ma, Michael E. Mussolino, Sankar D. Navaneethan, Amanda Marma Perak, Remy Poudel, Mary Rezk-Hanna, Gregory A. Roth, Emily B. Schroeder, Svati H. Shah, Evan L. Thacker, Lisa B. VanWagner, Salim S. Virani, Jenifer H. Voecks, Nae-Yuh Wang, Kristine Yaffe, Seth S. Martin, and null null. Heart disease and stroke statistics—2022 update: A report from the american heart association. *Circulation*, 145(8):e153–e639, 2022.

[4] Jelmer M. Wolterink, Robbert W. van Hamersvelt, Max A. Viergever, Tim Leiner, and Ivana Išgum. Coronary artery centerline extraction in cardiac ct angiography using a cnn-based orientation classifier. *Medical Image Analysis*, 51:46–60, 2019.

[5] Majd Zreik, Nils Hampe, Tim Leiner, Nadih Khalili, Jelmer M. Wolterink, Michiel Voskuil, Max A. Viergever, and Ivana Išgum. Combined analysis of coronary arteries and the left ventricular myocardium in cardiac ct angiography for detection of patients with functionally significant stenosis. In Ivana Išgum and Bennett A. Landman, editors, *Medical Imaging 2021: Image Processing*, volume 11596, pages 394 – 401. International Society for Optics and Photonics, SPIE, 2021.