

Deep Convolutional Neural Networks for Real-Time Patient-Specific Wall Shear Stress Estimation

Chris Mamon, Eric P. Wong, Eric K. W. Poon, Vikas Thondapu, Peter Barlis, Andrew Ooi and Rao Kotagiri

Abstract—Blood flow exerts stresses on the coronary artery wall. Wall shear stress (WSS), the parallel component of stress, when abnormally low ($< 1\text{Pa}$) may promote the pathological development of cholesterol-laden plaque, resulting in arterial narrowing. Accurately determining WSS helps to inform clinicians about plaque progression. Nevertheless, WSS is difficult to measure directly but can be estimated accurately by solving the governing equations of fluid motion (the Navier-Stokes Equations). Solving these complex equations can be both time consuming and computationally expensive to perform ($> 1\text{hr}$). Deep Learning (DL) is capable of learning patterns directly from data providing an avenue for real-time prediction of WSS. Thus, in this paper, we outline a method which employs popular DL models for their use in the prediction of WSS from artery models generated from coronary angiography. The best performing model was able to predict the WSS in under 0.29s with an average coefficient of determination $R^2 > 75\%$ compared to high resolution CFD simulations.

Index Terms—Computational Fluid Dynamics, Coronary Angiography, Coronary Artery Disease, Deep Learning, Machine Learning.

I. INTRODUCTION

THE dynamic motion of blood flowing through the coronary artery exerts hemodynamic forces on the vessel wall. These forces have both direction and magnitude and

C. Mamon is with the Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: cmamon@student.unimelb.edu.au)

E. P. Wong is with the Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: ericw2@student.unimelb.edu.au)

E. K. W. Poon is with the Department of Medicine, Faculty of Medicine, Dentistry and Health Sciences, Melbourne Medical School, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: epoon@unimelb.edu.au)

Peter Barlis is with the Department of Medicine, Faculty of Medicine, Dentistry and Health Sciences, Melbourne Medical School, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: peter.barlis@unimelb.edu.au)

V. Thondapu is with the Massachusetts General Hospital, Harvard Medical School Cardiovascular Imaging Research Center Boston, Massachusetts, (e-mail: vthondapu@mgh.harvard.edu)

Andrew Ooi is with the Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: a.ooi@unimelb.edu.au)

K. Ramamohanarao is with the Department of Computing and Information Systems, Melbourne School of Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, 3010 Victoria, Australia, (e-mail: rkotagiri@gmail.com)

Manuscript received June 19, 2020

can be broken down into components perpendicular (normal) and tangential to the wall. To quantify the forces experienced by the cells the force components are divided by the acting surface area resulting in the wall stresses (measured in $\text{Pa} = \text{dynes}/\text{cm}^2$). The perpendicular component of stress, pressure, is associated with the deformation of the cells within the wall while the tangential component of stress, wall shear stress (WSS), exerts a shearing deformation along the surface. This shearing deformation is sensed by endothelial cells lining the interface between the wall and blood via stress-responsive cell components [1]. This interaction regulates the endothelial cell's structure and function by activating or deactivating intracellular pathways, thereby playing a continuous role in arterial homeostasis and pathology.

Through analysis of the magnitude and variations of the WSS it has been observed that when WSS is high or unidirectional ($> 7\text{Pa}$) the endothelial cells presume a dormant state, while low and/or directional WSS ($< 1\text{Pa}$) is pro-inflammatory, promoting the infiltration of fatty deposits into the arterial wall [2]–[4]. Stone et al [5] show that plaque burden complemented by low WSS predicted further plaque growth. This plaque growth contributes to arterial narrowing, a defining characteristic of ischemic heart disease (IHD). Continued plaque growth can culminate in plaque rupture and heart attacks [6]. As such, IHD remains the leading cause of death globally. Calculation and study of WSS, therefore, has the potential to inform our understanding of IHD and impact clinical decision making.

The WSS along the artery depends highly on the geometry of the vessel and can therefore vary significantly along the artery. Specifically, variations of the surface curvature along the artery are highly correlated with changes in the WSS [7], [8]. For example regions which curve internally along the artery tend to form pockets of physiologically low WSS ($< 1\text{Pa}$), whilst regions where the surface is locally elevated tend to form high WSS ($> 7\text{Pa}$) [1].

Nonetheless, WSS, cannot be measured directly in the coronary arteries, thus, it has been computed over models of the artery by solving the governing equations of fluid motion, the Navier-Stokes equations [1], [9]. Solutions to these complex equations are generally estimated through computational fluid dynamics (CFD), via the finite volume method (FVM). In FVM, the domain, i.e. artery, is subdivided into many small volumetric elements over which the Navier-Stokes equations can be simplified and solved locally [10]. This process is per-

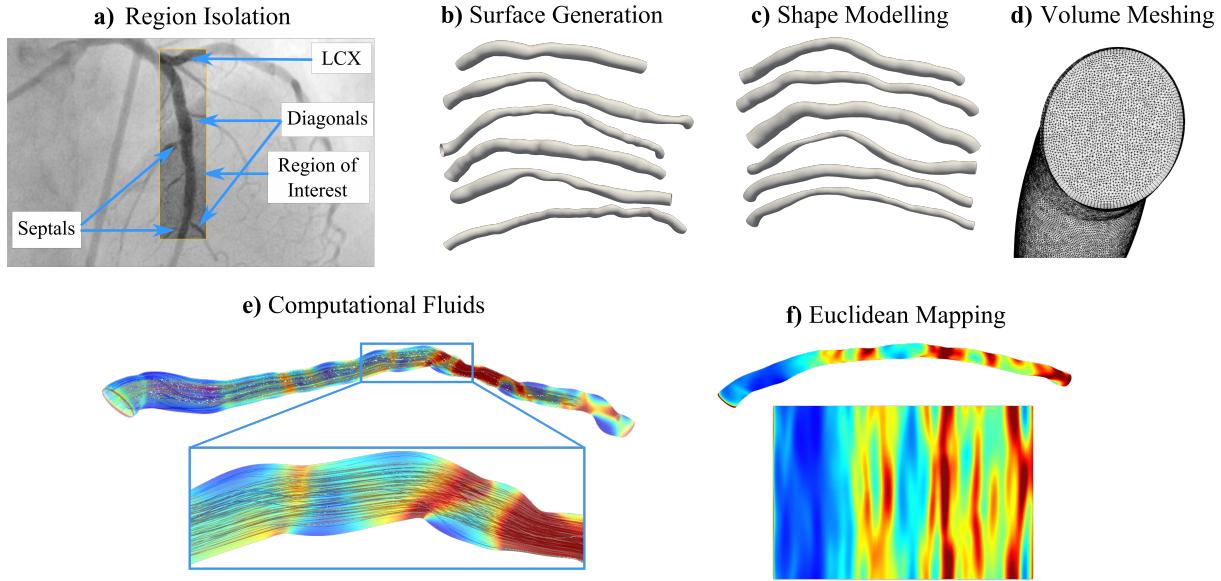


Fig. 1. a) shows the region of interest selected and the landmarks used to identify the region: Left circumflex (LCX), diagonals and septals b) is an example of a patient-specific surface generated from angiography a), c) shows a sample of artificial geometries generated from the statistical shape models, d) shows a volume mesh generated from the surfaces in c), e) shows an example CFD solution f) is a mapping of the WSS along the artery surface to a 2-D regular grid

formed over multiple discretised and sequential time instances allowing the modelling of the flow's chronological and spatial evolution. However, accuracy and stability of the resulting solution is highly dependent on the spatial and temporal refinement of such elements, thus, the artery is divided into hundreds of thousands to millions of volumetric elements and hundreds to thousands of discrete-time instances. Because of this, CFD can be expensive due to the necessary hardware, expertise and time required to complete a simulation (up to 6 hours or more for complex cases even with high-performance cloud computing) [11]–[13].

Machine learning (ML) can be used to approximate CFD simulation by leveraging knowledge gleaned from past simulations. Such recent examples include ML for the approximation of fractional flow reserve, complex partial differential equations and aortic wall stresses [14]–[21]. Such methods displayed upwards of 100x speedup over traditional numerical methods. Also once trained, these ML models were capable of performing predictions on standard workstations. Thus, ML models trained on WSS have the potential to reduce the barrier of entry for clinical CFD and can act as both complementary diagnostic tools or an input feature for an ML diagnostic method. We explore for, WSS prediction, deep learning (DL) convolutional neural network (CNN) models [22]. Our models are trained and tested for the prediction of WSS from CFD ground truths. CNN's are an extremely popular DL architecture inspired by the structure of the visual system [23]. As a variety of CNN architectures are used in computer vision this work explores the effectiveness of popular CNN architectures for WSS prediction. Specifically, we explore encoder-decoder fully convolutional neural network (FCNN) architectures such as the U-Net for WSS prediction [24].

U-Net like architectures are prominent for a wide range

of tasks since they can deal with arbitrarily sized inputs and generalize well in lieu of large datasets, [24]. U-Nets also tend to be symmetric, outputting features of the same size as the input. Combined with skip connections [25] they can resolve high level, low resolution features with low level, high resolution features, allowing such networks to predict detailed localized features necessary for CFD application. With this in mind we explore the following popular architectures; Unet [24], LinkNet [26] and Feature Pyramid Net [27]. These encoder-decoder architectures are also highly modular, thus, we explore Visual Geometry Group (VGG) [28] vs Resnet [29] encoders with added squeeze and excitation modules [30], [31].

To summarize, in this work we describe a method which predicts the magnitude of WSS (MWSS) along a region of interest along the coronary artery using CNN's for patient specific artery models generated from coronary angiography, where the WSS was obtained from high-resolution CFD. The paper is organized as follows: the materials and methods are described in Section II; Section III presents the results and finally, Section IV presents a discussion as well as the drawn conclusions.

II. MATERIALS AND METHOD

The CNN was used to approximate $f(q(x)) = g(x) = MWSS$ where f is an unknown function which maps the flow rate, q , to the $MWSS$ where x is some feature describing the surface. To generate training and testing examples we initially isolated a region of interest along patient coronary angiography. We chosen to focus on the proximal LAD because lesions occurring here are associated with higher mortality [32]. Consistent anatomic boundaries were used to define a consistent region between all patients: the left circumflex

(LCX) artery and the second septal were chosen as the inlet and outlet landmarks respectively. We supplemented our sample geometries with artificial geometries generated from statistical shape modelling (SSM) [33] similar to Liang *et al* [20]. To allow for the prediction of the MWSS by the CNN the surfaces were mapped to structured grids. To simulate varying inlet flow rates a scaling law was applied to each vessel based on an inlet diameter flow relationship [34]. In general, hemodynamic parameters are time averaged during analysis, in this work we employed steady flow conditions, however, the methods are extendable to unsteady flow conditions, where the output is the desired time-averaged hemodynamic parameter. A summary of the procedure to generate the CNN training and testing samples is shown in figure 1.

A total of 17 patient-specific arteries were reconstructed from which 284 phantom arteries were generated. Each surface consisted of approximately 98,000 surface points. Our experiments were divided into two stages: model selection and patient-specific testing. For model selection, training and testing was performed on purely SSM models. The best performing model was passed to the second stage, during which, the network was trained on all the SSM models and a subset of patient models before being tested on a holdout set of purely patient data.

Deep learning models were generated with PyTorch 1.30 (Facebook AI Research, Long Beach, CA, USA) [35]. Code was implemented in Python 3.7 (Python Software Foundation) and trained on an Intel Core i9 7900X @ 3.30GHz CPU, 64GB ram with a NVIDIA GeForce GTX 1080 GPU.

A. Patient Data and Model Reconstruction

In total, we started off with 17 patient coronary angiographies from a previous randomized controlled trial conducted by our group (NCT01776567) with imaging suitable for 3-dimensional reconstruction and CFD analysis. All angiography surfaces were output from QAngio XA 3D RE, (Medis, Leiden, the Netherlands) as a point cloud consisting of a set of ellipses in 3D. To establish point correspondence between different patients and to facilitate the SSM, the set of elliptical curves are converted into stereolithography (STL) surfaces with the same number of nodes and the same connectivity among the elements for all patients. The output of QAngio was a rough STL surface, which was unsuitable for CFD. Thus, the surfaces are initially smoothed by mapping the surface to a 2D rectangle and applying a 2D Savitzky-Golay filter to the surface with constant padding in the longitudinal direction and cyclic padding in the azimuthal. From the grid, a surface was then fit with non-uniform rational B-splines (NURBS).

B. Statistical Shape Modelling

Once the surfaces are generated, to create the SSM, they must be aligned to a common co-ordinate system. Each surface can be thought of as a shape composed of a set of points. These shapes can be aligned by applying an iterative approach, specifically Generalized Procrustes Analysis (GPA) [36].

For the following explanation superscripts specify shapes while subscripts describe shape components. We assume that

the set of shapes $\{\Gamma^1, \dots, \Gamma^k\}$, where k is the total number of shapes, are in correspondence such that for a reference shape Γ^R and a selected shape Γ^C each point (which is a vector) $\mathbf{x} \in \Gamma^R$ a point $\mathbf{x}' \in \Gamma^C$ exists in correspondence. For the two shapes we can eliminate the effects of rotation and translation by minimizing the following cost function,

$$(t, R) = \operatorname{argmin}_{(t, R)} \sum_{i=1}^n |\mathbf{x}_i - \mathbf{R}(\mathbf{x}'_i - t)|^2, \quad t \in \mathbf{R}^3, \mathbf{R} \in \mathbf{R}^{3 \times 3} \quad (1)$$

Where t and \mathbf{R} are the translation and rotation that superimposes the two shapes and n is the total number of points per shape. The solution to the optimization problem can be computed efficiently in 3D with Kabsh's Algorithm [37].

Once the shapes are aligned, we seek a model which is parameterized and of the form $\Gamma = F(b)$ where b is a vector of the model's parameters. One method of obtaining this model is through principal component analysis (PCA) [38]. The method extracts principal components which describe the set of shapes. For a chosen shape Γ^C of n points in m dimensions, the shape can be represented by a nm element vector formed by concatenating the co-ordinate vectors. The following example shows the resulting vector in 3 dimensions of the shape Γ^C where its n -th point $\mathbf{x}_n = (x_n, y_n, z_n)$,

$$\Gamma^C = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n)^T \quad (2)$$

We convert our aligned shapes into element vector form. By stacking these vectors we obtain a matrix containing all shapes (shape matrix),

$$M = (\Gamma^1, \Gamma^2, \dots, \Gamma^k) \quad (3)$$

To obtain the information necessary for our model we analyse the covariance matrix of the shape matrix,

$$C(M) = \frac{1}{k-1} \sum_{l=1}^k (\Gamma^l - \Gamma^\mu) (\Gamma^l - \Gamma^\mu)^T \quad (4)$$

where,

$$\Gamma^\mu = \frac{\sum_{i=1}^k \Gamma^i}{k} \quad (5)$$

By analysing this covariance matrix and obtaining its eigenvalues and eigenvectors we observe that a new shape can be modelled through linear combination of the mean shape and the calculated eigenvalues and eigenvectors (also known as modes),

$$\Gamma = \Gamma^\mu + \sum_{i=1}^p a_i \sqrt{\lambda^i} V^i \quad (6)$$

Where the set $\{\lambda^1, \dots, \lambda^i, \dots, \lambda^p\}$ and $\{V^1, \dots, V^i, \dots, V^p\}$ are the eigenvalues and eigenvectors of C sorted from largest to smallest where p represents the total number of selected modes and a_i represents a modifiable parameter $-3 \leq a_i \leq 3$ (i.e. the shape deformation along the chosen principle axis was within 3 standard deviations

of the mean). As the covariance matrix is quite large these eigenvalues and eigenvectors can be obtained via singular valued decomposition of the deformation matrix,

$$D = M - \Gamma^\mu \quad (7)$$

Shape modes were selected such that they explain a large percentage of the total shape variation.

$$\frac{\text{Explained Variation}}{\text{Total Variation}} = \frac{\sum_{i=1}^p \lambda_i^p}{\sum_{i=1}^K \lambda_i^k} \quad (8)$$

For our study the value $p = 11$ was chosen to explain 90% of the total shape variation. Statistics of the shape mode is shown in table I.

TABLE I

STATISTICS FOR THE PHANTOM SHAPES GENERATED BY THE SSM

Parameter	Mean (Standard Deviation)
Inlet Area (mm^2)	9.017 (2.642)
Inlet Diameter (mm)	3.352 (0.498)
Outlet Area (mm^2)	3.881 (1.076)
Outlet Diameter (mm)	2.201 (0.311)
Minimum Lumen Area (mm^2)	3.407 (1.170)
Minimum Lumen Diameter (mm)	2.051 (0.362)
Centerline Length (mm)	59.409 (0.545)

C. Surface Mesh Parameterization

The unwrapping of our artery to a Euclidean grid is essentially a re-meshing procedure over a fixed border parameterization. Commonly, such a remeshing procedure involves three stages. Initially, a border for the parameterization is decided. Afterwards, a piecewise linear mapping, Ψ , is constructed which maps a piecewise linear mesh patch M of our mesh to an isomorphic planar triangulation $U \in \text{IR}^2$. The mapping, Ψ , must best preserve the intrinsic properties of M . After Ψ is constructed the desired mesh can be generated by interpolating the properties of our original mesh to a set of structured vertices over U .

The high performance Computational Geometry Algorithms Library (CGAL) provides a set of methods which deals with meshes topologically equivalent (homeomorphic) to disks. Meshes with an arbitrary topology can be parameterized, provided, such a boundary is defined. The resulting disks can then be mapped onto a square given a set of corner points. For our artery this boundary can be defined as the set of edges comprised of our inlet and outlet boundaries, and the shortest cut graph between our inlet and outlet vertices (seam). To obtain the seam we apply the heat method [39], selecting the inlet vertices as the set of sources. From the distance field the closest outlet vertex can be selected. Applying Dijkstra's algorithm [40] on the selected vertex the seam can then be obtained as the shortest path from the inlet vertices to the outlet vertices. The corners of our parameterization are then defined as the vertices of our seam located at the inlet and outlet boundaries.

Once the boundary and the corners of our parameterization is defined the mapping can be performed via the discrete

authalic parameterization method introduced by Desbrun et al, [41]. The method locally minimizes the area distortion. After the mapping is completed re-meshing can then be performed by barycentrically interpolating, [42], the properties of M over U such that a rectangular grid is generated. We must note however that as a grid is used the distance between vertices when viewed by the CNN is assumed to have a weighting of 1. This leads to local distortions, nevertheless CNN's have been shown to be robust to noise and have been shown to provide highly correlated MWSS predictions.

D. Computational Fluid Dynamics

After the shapes are created, they must be re-meshed into a volume mesh for CFD. The remeshing is performed in Pointwise V18R2 (Pointwise Inc., Fort Worth, USA). The mesh used was an unstructured domain with anisotropic layers growing at the boundaries. Near wall layers of each artery were composed of quad meshes and the core of the artery was composed of tetrahedral elements. Each artery contained between 1 and 4 million cells.

The governing equations of fluid flow are the Navier-Stokes Equations,

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla P + \nabla \cdot (\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)) \quad (9)$$

$$\nabla \cdot \mathbf{u} = 0$$

High fidelity CFD simulations were performed using OpenFOAM 2.1.1 (OpenCFD Ltd, ESI group, Bracknell, UK) that directly solved the incompressible Navier-Stokes equations. The wall for all arteries were assumed to be rigid, and the flow conditions were modelled as Newtonian with a dynamic viscosity $\nu = 3.302 \times 10^{-6} \text{kg/m/s}$. A representative inlet flow rate was applied to each case using a diameter flow rate relationship [34],

$$q_{in} = 1.43d^{2.55} \quad (10)$$

Where the inlet flow rate, q_{in} , is calculated from the area assumed diameter, d , of the artery inlet. Each case was run until steady state was achieved, of which, took approximately 2 to 4 hours using a single 16 core Intel Xeon Gold 6130, 2.1GHz Skylake processor, from the National Computation Infrastructure's High-Performance Computing environment, Raijin¹. Multiple simulations were run in parallel, of which, were managed automatically via Python. After the CFD was completed the WSS vector was computed,

$$WSS = \tau_t - (\tau_t \cdot \mathbf{n}) \mathbf{n} \quad (11)$$

Where τ is the wall traction vector calculated from the stress tensor and \mathbf{n} is the surface normal unit vector [43]. For the final analysis the magnitude, MWSS, of the WSS vector is taken.

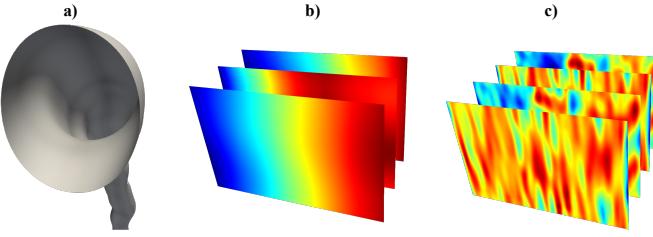


Fig. 2. We show a) an artery surface where the b) point features and c) curvature features have been mapped to a regular grid

E. Feature Input

The choice of x is a critical aspect of model development, as careful selection can significantly improve performance (i.e. generalization) and training convergence [44]. The set of points, 2 b), which compose the artery's surface can be used as x , however, such points can vary in magnitude consistently, potentially reducing CNN performance, as they depend on both artery size and location. Curvature describes the amount by which a curve deviates from a straight line. Unlike a field of points, the magnitude of curvature, 2 b), depends on the space local to a point and is independent of the location of the points in space. Thus, curvature acts to normalize the training data. Combined with its strong link with the evolution of WSS [7], [8], the feature was used to improve predictions.

F. Curvature

The curvature of a geometry can be thought of as the amount a curve deviates from a straight line. For a curve C at some point P this value is a single scalar quantity calculated from circle which fits P and its infinitesimally nearby points (osculating circle). Along a surface embedded in Euclidean space the curvature must be periodic, giving rise to Maximal, Minimal, Mean and Gaussian curvature. Given a plane F containing the surface normal n at some point P of a parameterized curve $K : r = r(x^1(t), x^2(t)) = r(t) = (r_1(t), r_2(t))$ along a regular surface S , the curvature of the curve of intersection between F and S (the normal section), C_F , is given at P by,

$$\kappa_F = \frac{II}{I} \quad (12)$$

Where I is the First Fundamental Form (FFF) of C , also known as the arc length along C ,

$$I = \left(\frac{ds}{dt} \right)^2 = \|\dot{r}\|^2 = \dot{r} \cdot \dot{r} = \sum_{i=1}^2 \sum_{j=1}^2 r_i r_j \frac{dx^i}{dt} \frac{dx^j}{dt} \quad (13)$$

While II are the symmetric coefficients of the Second Fundamental Form (SFF) of S ,

$$II = \sum_{i=1}^2 \sum_{j=1}^2 n \frac{\delta^2 r}{\delta x^i \delta x^j} dx^i dx^j \quad (14)$$

¹<https://opus.nci.org.au/display/Help/Raijin+User+Guide>

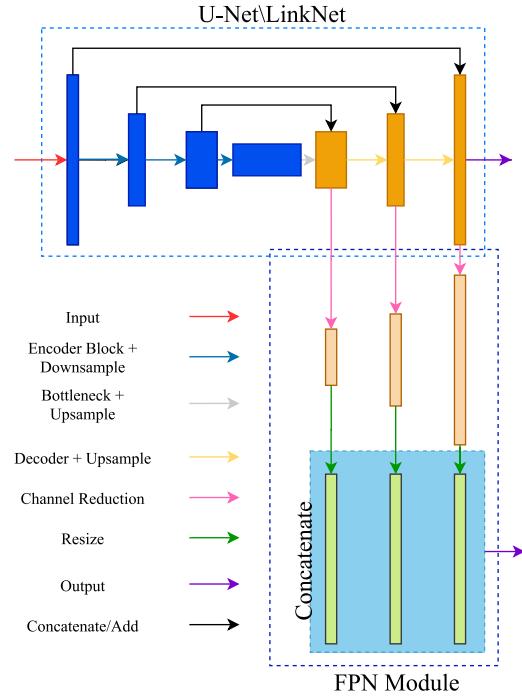


Fig. 3. General overview of DL architectures. The FPN module can be attached directly to the decoder of the desired model. For such cases the prediction was made via the FPN module

By rotating the plane F about n the curvature κ_F around C_F is calculated and we observe that the curvature around P is periodic. The maximum and minimum of this osculating curvature will be defined, respectively as,

$$\max \kappa_F = \kappa_1 \quad \min \kappa_F = \kappa_2 \quad (15)$$

These two values of curvature are called the principal curves, while their directions are the principal directions. By taking the product and the sum of the principal curvatures we obtain the Gaussian and Mean curvatures,

$$K = \kappa_1 \kappa_2 \quad H = \frac{1}{2} (\kappa_1 + \kappa_2) \quad (16)$$

Surfaces are treated as discrete where the curvatures were calculated through the fitting of normal section curvatures, described by Zhang *et al.* [45].

G. Deep Learning Architecture

A general overview of the architectures used is described in figure 3. In general a U-Net/LinkNet is an encoder decoder architecture which uses either addition or concatenation to recover high level features. An FPN module essentially collates features of various scales by reducing, upsampling and concatenating the output maps of the decoder. The FPN module can be directly attached to the decoder of the U-Net/LinkNet. Further details about the architectures can be found directly from the papers outlined in the introduction.

For the U-Net and many models the encoder building block is Conv+ReLU or Conv+BNorm+ReLU and decoding is performed via transposed convolutions. Instance Normalization has been used to improve and speed up the convergence of

TABLE II

5FCV OF TESTED MODELS. TOTAL NUMBER OF PARAMETERS PER MODEL ARE SHOWN IN MILLIONS. AVERAGE TRAINING TIME PER FOLD IS SHOWN IN HOURS:MINUTES:SECONDS. ACCURACY IS PRESENTED IN % MEAN (STANDARD DEVIATION). THE BEST-PERFORMING MODEL IS HIGHLIGHTED IN BOLD. (HIGHER MEAN AND LOWER STANDARD DEVIATION BETTER)

Model	Parameters Millions	Training Time hrs:min:sec	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Expected
U-Net	34.521	01:25:35	93.684 (4.203)	92.244 (4.906)	91.221 (10.264)	92.602 (6.245)	93.240 (6.541)	92.598 (6.432)
RN34+U-Net	20.293	01:58:43	92.848 (4.756)	93.428 (5.528)	91.162 (8.537)	92.478 (6.586)	92.981 (7.003)	92.579 (6.482)
LinkNet	6.635	01:01:48	91.892 (4.710)	91.166 (5.398)	89.020 (9.288)	90.704 (9.716)	91.693 (6.530)	90.895 (7.128)
U+LinkNet	20.157	00:42:10	92.651 (4.163)	92.105 (5.430)	90.748 (9.244)	91.518 (8.771)	92.557 (6.281)	91.916 (6.778)
RN34+LinkNet	16.743	01:11:25	92.400 (4.400)	92.254 (6.316)	90.513 (7.969)	90.722 (8.485)	91.700 (6.530)	91.518 (6.740)
U+FPN	21.568	00:48:19	93.887 (3.416)	93.247 (5.817)	91.829 (8.207)	93.090 (6.436)	93.172 (6.162)	93.045 (6.007)
RN34+FPN	18.435	01:24:37	90.197 (5.195)	88.274 (7.299)	86.316 (11.202)	88.987 (6.315)	90.376 (5.044)	88.830 (7.011)
U+FPN+sSE	21.743	01:26:42	93.293 (4.206)	93.746 (5.546)	92.055 (8.090)	92.851 (6.871)	92.480 (5.736)	92.885 (6.090)
U+FPN+cSE	21.570	01:26:55	93.914 (3.751)	93.742 (5.075)	91.456 (8.927)	92.659 (7.861)	92.293 (6.674)	92.813 (6.458)
U+FPN+scSE	21.745	01:29:56	93.654 (4.109)	93.800 (5.149)	91.844 (8.929)	93.368 (6.219)	92.904 (6.228)	93.114 (6.127)

general adversarial networks (GANs) [46], whilst upsampling convolutions have been used to remove “checkerboard artefacts” from transposed convolutions [47]. Thus, we apply Conv+InstanceNorm+ReLU as the basic building block for our model encoders and use upsampling convolutions during decoding portions for both Resnet and U-Net. For Resnet encoders pre-activation was applied, as pre-activation can reduce overfitting and shows improved performance over the traditional post-activation method [48]. For all Resnet encoders the initial strided convolution and max pooling layer were removed. For all networks a ReLU is applied to the output layer to ensure that output values are strictly positive. For the best performing network we applied various squeeze and excitation modules; Spatial Squeeze and Channel Excitation (+sSE), Spatial Squeeze and Channel Excitation (+cSE) or Concurrent Spatial and Channel Squeeze and Channel Excitation (+scSE).

H. Training and Testing

In this work model evaluation for both stages was performed via 5-fold cross-validation (5FCV). The data was split into 0.8/0.2 train and test sets for each fold. Over fitting was mitigated by further dividing the training set into a 0.9/0.1 train and validation sets (thus, each fold is a data set with a 0.72/0.08/0.2 train, validation and test split). An epoch describes a pass through the entire training set. As the model may not have been trained after a single epoch, training was run for 100 epochs. All outputs of the network are ensured to be ≥ 0 as the output layer for all networks was passed through a ReLU. Thus, we define the metric for our regression accuracy as,

$$\text{accuracy} = \frac{1}{n} \sum_{i=0}^n \left(1 - 2 \frac{|1 - a_i|}{|1 + a_i|} \right) \quad (17)$$

for $\hat{y}_i = a_i y_i$, where $a_i \geq 0$, for the $i - \text{th}$ element our ground truth y and our prediction \hat{y} . After each epoch the mean of this accuracy was then computed over all arteries, after which the model with the highest mean accuracy was applied to the test set. The data was augmented at training time by rolling the input along its height by a randomly generated integer between 0 and 256. For both the following experiments the data was divided as described above. In the first experiment

only the SSM was used for 5FCV training and testing. In the second experiment, the model was trained on all SSM arteries and a subset of patient arteries models before being tested on patient only holdout sets.

The CNN's were trained with the L1-Loss, also known as the mean absolute error (MAE),

$$l = \sum_{i=0}^n |y_i - \hat{y}_i| \quad (18)$$

where the loss l is the sum of the difference between the $i - \text{th}$ element of our ground truth y and our prediction \hat{y} . All neural networks were optimized with Adam Optimization [49]. The learning rate (LR) was found through the LR finder algorithm developed by Smith et al. [50]. The LR suggested was the LR that generated the steepest sloped negative loss gradient. During the training procedure if the validation accuracy plateaued for 5 epochs the learning rate was reduced by a factor of 0.1.

III. RESULTS

Metrics resulting from the 5-fold cross-validation for all networks is shown in table II. Most models were capable of predicting the MWSS of the SSM with above 90% accuracy over the 5-folds. Nevertheless, we observed decreased performance when comparing the LinkNet to the base U-Net model, -1.703 (+0.646) %. This occurred despite using the same encoder. This indicated that feature bottlenecking reduced the performance of the network and that the aggregated information over all channels from full convolutions was important for accurate predictions.

Surprisingly, residual connections also reduced performance of the networks. This was consistent for all networks. However, it was most exacerbated when combined with the FCN Network which scored significantly lower than the U+FCN, +4.215 (-1.004).

The FCN models were the best performing models, highlighting the importance of contextual information learned over various scales via the feature pyramid. However, isolated attention from sSE and cSE modules decreased the performance. Nevertheless, when combined, i.e. U+FCN+scSE, displayed a minor increase in accuracy. As the U+FCN+scSE performed the best it was selected for further tests.

TABLE III

SUMMARY OF THE REGRESSION METRICS; ACCURACY, R^2 , RMSE AS WELL AS PIXEL-WISE CLASSIFICATION METRICS FOR VARIOUS THRESHOLDS OF MWSS SHOWN IN % MEAN (STANDARD DEVIATION), OF THE BEST PERFORMING CNN FOR BOTH TESTS ON SSM AND PURELY PATIENT SPECIFIC ARTERIES. THE RESULTS SHOWN ARE THE AVERAGE OF THE 5FCV. (HIGHER MEAN BETTER LOWER STANDARD DEVIATION BETTER, FOR RMSE LOWER MEAN IS BETTER)

	Coverage	F1-Score	Precision	Sensitivity	Specificity		Accuracy	r^2	RMSE
SSM									
MWSS ≤ 1	44.873 (16.380)	95.524 (2.838)	95.803 (4.535)	95.511 (3.860)	96.909 (3.118)				
MWSS ≤ 0.75	25.479 (13.277)	91.732 (5.475)	92.317 (8.411)	92.097 (6.484)	97.956 (2.056)		93.654 (4.109)	95.996 (1.722)	0.056 (0.032)
MWSS ≤ 0.5	5.323 (6.297)	72.764 (25.491)	75.015 (28.291)	77.106 (25.156)	99.340 (0.778)				
Patient Specific									
MWSS ≤ 1	41.638 (13.876)	83.552 (7.356)	86.018 (11.231)	84.291 (5.384)	91.677 (7.534)				
MWSS ≤ 0.75	27.967 (15.392)	78.351 (7.608)	85.178 (13.650)	77.065 (7.902)	95.934 (4.033)		76.259 (10.502)	76.026 (15.031)	0.223 (0.160)
MWSS ≤ 0.5	12.761 (11.017)	62.054 (16.041)	79.112 (16.756)	57.774 (20.795)	98.981 (1.095)				

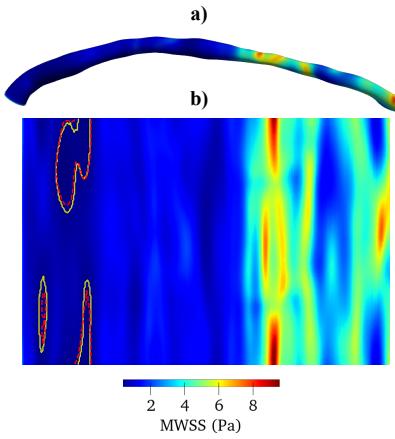


Fig. 4. Example low MWSS regions a) surface b) pixel-wise showing the contours for the CFD (yellow solid line) and CNN (red broken line) where the MWSS < 0.5 .

As it is important to know how well the pattern of the predicted MWSS matches with the ground truth as well as the potential amount of variation the coefficient of determination (R^2), [51], as well as the root mean-squared error (RMSE), [52], was calculated over the 5FCV and is shown for the SSM tests in table III.

Nevertheless, knowledge of the location of low MWSS is of clinical significance as the combination of low MWSS ($< 1\text{Pa}$) distal (away from the heart) to the throat of a region with large plaque burden has been shown to independently predict progressive enlargement of plaque and further artery narrowing [5]. Thus, we treat low MWSS identification as a pixel-wise classification problem by thresholding our predictions for the following ranges $< 1\text{Pa}$, $< 0.75\text{Pa}$, $< 0.5\text{Pa}$. An example of thresholded regions of low MWSS is shown in figure 4.

To grasp the predictive capabilities of the CNN for various levels of MWSS granularity we showed the amount by which each thresholded level of MWSS covered the surfaces as well as how precise, sensitive and specific the predictions were. Below we describe the metrics used.

Precision measures the total number of retrieved samples relevant to the query,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (19)$$

Sensitivity measures the fraction of the pixels within desired threshold successfully identified,

$$\text{Specificity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (20)$$

Specificity measures the fraction of pixels outside the desired threshold successfully identified,

$$\text{Sensitivity} = \frac{\text{TN}}{\text{TP} + \text{FN}} \quad (21)$$

Finally the F1-Score is a measure of accuracy which considers both the precision and the specificity,

$$\text{F1-Score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (22)$$

The average metrics over the 5FCV for pixel-wise classification of low MWSS regions when trained and tested on SSM is summarized in table III.

The final tests analysed the performance of the best CNN when re-applied to patient-specific models. The CNN model was pre-trained on a combined dataset of SSMs and the original patient-specific models. The CNN was then tested on purely patient-specific data over 5FCV. Table III shows the relative accuracy, coefficient of determination R^2 , RMSE and pixel-wise classification metrics. Examples of testing predictions made over the 5FCV are shown in figure 5. The top 4 predictions and the bottom 4 predictions are shown in descending order.

Figure 6 a) shows the linear regression of the natural log of sub-sampled MWSS predictions of CNN vs CFD. For each fold, for every artery 50 pixel locations were randomly sub-sampled and the regression was performed over the set of all the points per fold. The square of the Pearson's correlation coefficient r^2 , [53] was calculated and is also shown. Figure 6 b) shows the Bland-Altman plot of the CFD vs CNN predictions of MWSS. The scale was left as linear so that the absolute agreement rather than the relative agreement could be viewed. As the data was not normally distributed it was of more interest to analyse percentiles of the data. In the plot we showed the 95% and 50% limits of agreement.

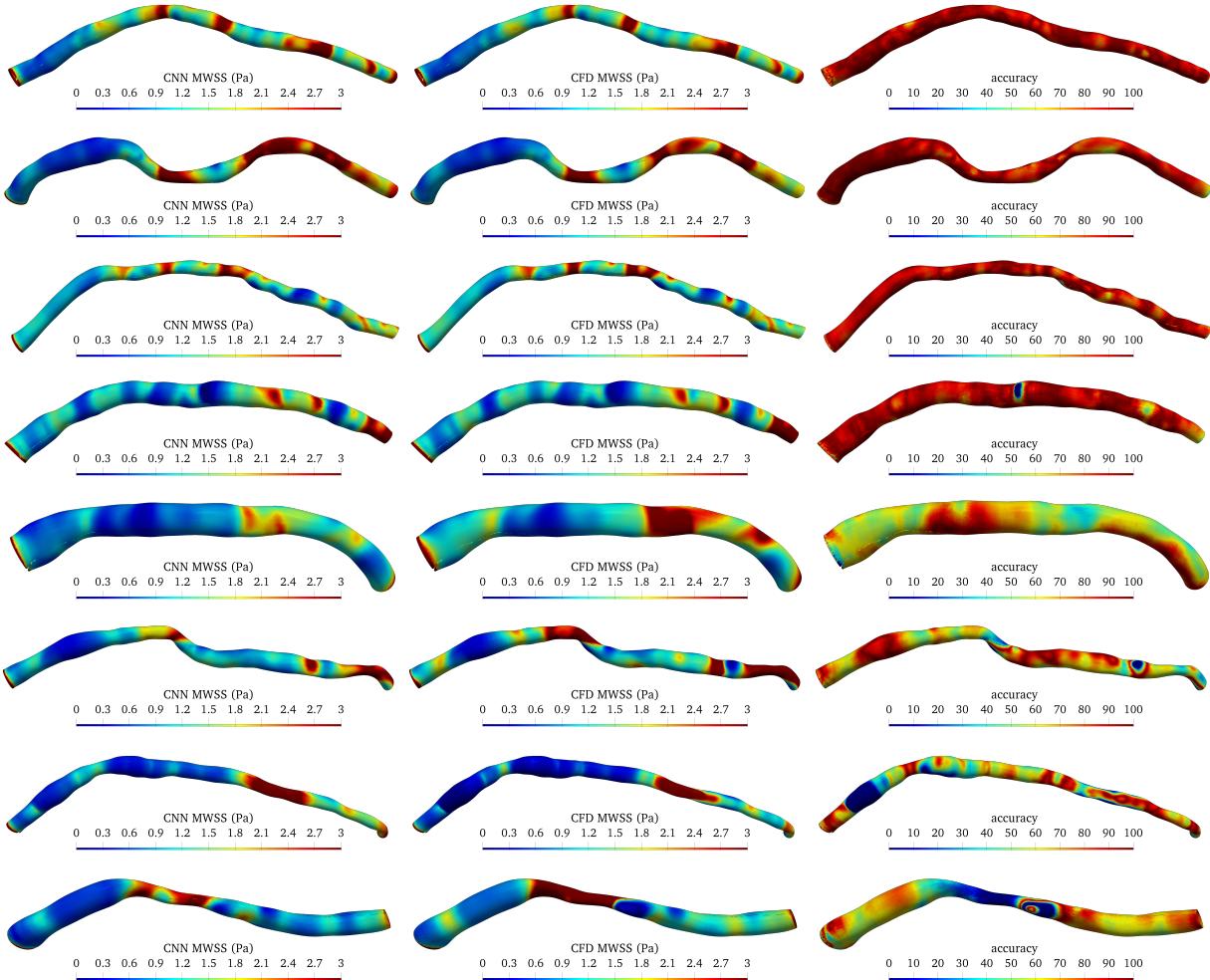


Fig. 5. Top 4 and bottom 4 predictions (in order) from the test sets of the 5FCV on the patient-specific models. The CNN and CFD predictions as well as well as pixel-wise relative accuracy is shown.

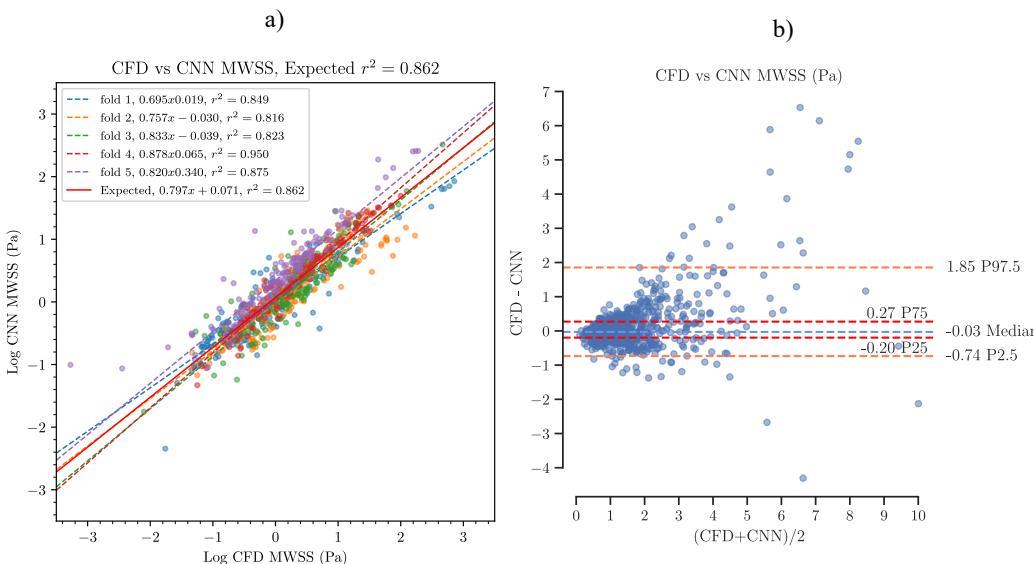


Fig. 6. a) linear regression of the log of the CNN vs the CFD predictions. The regression line and the Pearson's correlation coefficient r^2 for each fold is shown. b) shows Bland-Altman plot of the CNN vs CFD predictions is shown. The 95% and 50% limits of agreement are shown.

From figure 6 a) we observe a high degree of correlation, however, on average the slope of the regression was < 1 . This indicated that the CNN had a tendency to under approximate the MWSS. From figure 6 b) observed that the CNN tended to under approximate the MWSS to a more significant extent than under approximate.

The time taken for prediction of MWSS for 58 arteries was approximately 17 seconds, leading to an approximate processing time of 0.29s. This is significantly faster than CFD, which took approximately 2-4 hours per case.

IV. CONCLUSION

Knowledge of the spatial distribution of the MWSS has the potential to inform our understanding of IHD and impact clinical decision making. Nonetheless, prediction of MWSS using CFD can be expensive due to the necessary hardware, expertise and time required to complete a simulation. Here we described a novel method for isolating a clinical region of interest and predicting MWSS along the coronary artery for patient-specific boundary conditions using deep learning. We trained a CNN to make MWSS predictions which strongly replicates high resolution CFD, $R^2 > 90\%$ for SSM models and $R^2 > 75\%$ for patient models, with a comparable degree of accuracy $> 75\%$. When treated as a pixel-wise binary classification problem we showed that the CNN could sensitively, precisely and specifically locate regions of physiologically low MWSS, $> 80\%$ for all metrics. To the authors knowledge this is the first paper which explores deep learning as a tool for MWSS prediction in patient-specific arteries using patient-specific boundary conditions.

We note the small sample size of our patient-specific arteries and acknowledge the discrepancy in accuracy between the SSM predictions and the patient-specific predictions, 17.189%. Also, we observed from 6 the CNN's bias towards low MWSS. The discrepancy highlighted the need to improve the generation of artificial geometries to encompass a wider degree of variation and include more examples of high MWSS. Nevertheless, though we observe a courser level of prediction on patient-specific data, $\text{RMSE} \approx 0.2$, the low RMSE on the SSM data, $\text{RMSE} \approx 0.05$, indicates that CNN's have the potential to perform very detailed MWSS predictions given synthetic dataset generation is improved.

Despite a decent training time of 1-2 hours, once trained, a CNN can perform MWSS predictions in near real time and potentially on mobile applications. This suggests that though the initial cost of production of examples is high, both in time and hardware requirements, once weights for the network are generated, predictions can be performed in a clinic quickly with a moderate to high performance workstation or mobile device. Also, once trained, the CNN only requires point data rather than complex meshes, further reducing the time constraints needed for MWSS prediction. This makes it a simple, fast and accessible alternative to CFD.

This method can potentially predict a wide variety of hemodynamic parameters along the artery surface. Since the CNN can quickly visualize these parameters for a given geometry any effect of deformation along the artery on the

hemodynamics can be analysed and used to inform clinical decisions. Also, prediction of hemodynamic parameters can be combined with other clinical information and passed into a ML model. For example, a model can combine MWSS with cross sectional imaging information to locate potential regions where plaque could grow or intervention may be beneficial.

We note that we have not included the complex interactions of the various side-branches, mechanical properties of the artery, the Non-Newtonian nature of blood or the time varying nature of blood flow. Future research is necessary to address the degree by which these simplifying assumptions can be made. Nonetheless such simulations can provide useful information when combined with additional medical data such as plaque burden as shown by Stone *et al*, [5]. We also note a significantly larger data set with larger variations will be necessary to accurately model MWSS for a general population. To generate this volume of data, an efficient process for generating training examples is necessary. Also, methods which asses the similarity of the test data to the training data will be necessary to accurately confirm the confidence of the predictions. Future work will attempt to tackle these problems and create an optimal pipeline for DL based CFD assessment of the coronary arteries. Nonetheless we showed that DL models can accurately predict MWSS directly from patient geometry given patient specific-boundary conditions.

V. DATA AND CODE AVAILABILITY

Permission to access statistical shape model data can be obtained via direct request to the author. Due to privacy restrictions patient specific models will not be made available. Code is currently being developed and maintained on GitHub. Code used for training and testing the CNN can be found at <https://github.com/Chrlsc0de/myTorch>. Code used to generate the dataset can be found at <https://github.com/Chrlsc0de/pymethods>. Access to the code can be obtained from direct request to the author.

REFERENCES

- [1] F. Gijsen, Y. Katagiri, P. Barlis, *et al.*, "Expert recommendations on the assessment of wall shear stress in human coronary arteries: Existing methodologies, technical considerations, and clinical applications," *European Heart Journal*, vol. 40, no. 41, pp. 3421–3433, Sep. 2019.
- [2] B. R. Kwak, M. Bäck, M.-L. Bochaton-Piallat, *et al.*, "Biomechanical factors in atherosclerosis: Mechanisms and clinical implications," *European Heart Journal*, vol. 35, no. 43, pp. 3013–3020, Sep. 2014.
- [3] H. Samady, P. Eshtehardi, M. C. McDaniel, *et al.*, "Coronary artery wall shear stress is associated with progression and transformation of atherosclerotic plaque and arterial remodeling in patients with coronary artery disease," *Circulation*, vol. 124, no. 7, pp. 779–788, Aug. 2011.
- [4] V. Thondapu, C. V. Bourantas, N. Foin, *et al.*, "Biomechanical stress in coronary atherosclerosis: Emerging insights from computational modelling," *European Heart Journal*, ehv689, Feb. 2016.
- [5] P. H. Stone, S. Saito, S. Takahashi, *et al.*, "Prediction of progression of coronary artery disease and clinical outcomes using vascular profiling of endothelial shear stress and arterial plaque characteristics," *Circulation*, vol. 126, no. 2, pp. 172–181, Jul. 2012.
- [6] K. C. Koskinas, Y. S. Chatzizisis, A. B. Baker, *et al.*, "The role of low endothelial shear stress in the conversion of atherosclerotic lesions from stable to unstable plaque," *Current Opinion in Cardiology*, vol. 24, no. 6, pp. 580–590, Nov. 2009.
- [7] V. A. Nosovitsky, O. J. Illegbusi, J. Jiang, *et al.*, "Effects of curvature and stenosis-like narrowing on wall shear stress in a coronary artery model with phasic flow," *Computers and Biomedical Research*, vol. 30, no. 1, pp. 61–82, Feb. 1997.

- [8] A. Wahle, J. J. Lopez, M. E. Olszewski, *et al.*, "Plaque development, vessel curvature, and wall shear stress in coronary arteries assessed by x-ray angiography and intravascular ultrasound," *Medical Image Analysis*, vol. 10, no. 4, pp. 615–631, Aug. 2006.
- [9] V. Thondapu, E. Tenekecioglu, E. K. W. Poon, *et al.*, "Endothelial shear stress 5 years after implantation of a coronary bioresorbable scaffold," *European Heart Journal*, vol. 39, no. 18, pp. 1602–1609, Feb. 2018.
- [10] F. Moukalled, L. Mangani, and M. Darwish, *The Finite Volume Method in Computational Fluid Dynamics*. Springer International Publishing, 2016.
- [11] L. Zhong, J.-M. Zhang, B. Su, *et al.*, "Application of patient-specific computational fluid dynamics in coronary and intra-cardiac flow simulations: Challenges and opportunities," *Frontiers in Physiology*, vol. 9, Jun. 2018.
- [12] P. D. Morris, A. Narracott, H. von Tengg-Kobligk, *et al.*, "Computational fluid dynamics modelling in cardiovascular medicine," *Heart*, vol. 102, no. 1, pp. 18–28, Oct. 2015.
- [13] C. Chiastra, S. Migliori, F. Burzotta, *et al.*, "Patient-specific modeling of stented coronary arteries reconstructed from optical coherence tomography: Towards a widespread clinical use of fluid dynamics analyses," *Journal of Cardiovascular Translational Research*, vol. 11, no. 2, pp. 156–172, Dec. 2017.
- [14] H. Cho, J.-G. Lee, S.-J. Kang, *et al.*, "Angiography-based machine learning for predicting fractional flow reserve in intermediate coronary artery lesions," *Journal of the American Heart Association*, vol. 8, no. 4, Feb. 2019.
- [15] O. Hennigh, "Lat-net: Compressing lattice boltzmann flow simulations using deep neural networks," May 25, 2017. arXiv: <http://arxiv.org/abs/1705.09036v1> [stat.ML].
- [16] J. Na, K. Jeon, and W. B. Lee, "Toxic gas release modeling for real-time analysis using variational autoencoder with convolutional neural networks," *Chemical Engineering Science*, vol. 181, pp. 68–78, May 2018.
- [17] L. Liang, W. Mao, and W. Sun, "A feasibility study of deep learning for predicting hemodynamics of human thoracic aorta," *Journal of Biomechanics*, vol. 99, p. 109544, Jan. 2020.
- [18] M. Jordanski, M. Radovic, Z. Milosevic, *et al.*, "Machine learning approach for predicting wall shear distribution for abdominal aortic aneurysm and carotid bifurcation models," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 537–544, Mar. 2018.
- [19] Z. Nie, H. Jiang, and L. B. Kara, "Stress field prediction in cantilevered structures using convolutional neural networks," Aug. 27, 2018. arXiv: <http://arxiv.org/abs/1808.08914v3> [cs.LG].
- [20] L. Liang, M. Liu, C. Martin, and W. Sun, "A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis," *Journal of The Royal Society Interface*, vol. 15, no. 138, p. 20170844, Jan. 2018.
- [21] T. Nishimichi, M. Takada, R. Takahashi, *et al.*, "Dark quest. i. fast and accurate emulation of halo clustering statistics and its application to galaxy clustering," *The Astrophysical Journal*, vol. 884, no. 1, p. 29, Oct. 2019.
- [22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [23] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, Feb. 2018.
- [24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," May 18, 2015. arXiv: <http://arxiv.org/abs/1505.04597v1> [cs.CV].
- [25] M. Drozdzal, E. Vorontsov, G. Chartrand, *et al.*, "The importance of skip connections in biomedical image segmentation," Aug. 14, 2016. arXiv: <http://arxiv.org/abs/1608.04117v2> [cs.CV].
- [26] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," Jun. 14, 2017. arXiv: <http://arxiv.org/abs/1707.03718v1> [cs.CV].
- [27] T.-Y. Lin, P. Dollár, R. Girshick, *et al.*, "Feature pyramid networks for object detection," Dec. 9, 2016. arXiv: <http://arxiv.org/abs/1612.03144v2> [cs.CV].
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 4, 2014. arXiv: <http://arxiv.org/abs/1409.1556v6> [cs.CV].
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 10, 2015. arXiv: <http://arxiv.org/abs/1512.03385v1> [cs.CV].
- [30] J. Hu, L. Shen, S. Albanie, *et al.*, "Squeeze-and-excitation networks," Sep. 5, 2017. arXiv: <http://arxiv.org/abs/1709.01507v4> [cs.CV].
- [31] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel squeeze & excitation in fully convolutional networks," Mar. 7, 2018. arXiv: <http://arxiv.org/abs/1803.02579v2> [cs.CV].
- [32] R. M. Califf, Y. Tomabechi, K. L. Lee, *et al.*, "Outcome in one-vessel coronary artery disease," *Circulation*, vol. 67, no. 2, pp. 283–290, Feb. 1983.
- [33] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995.
- [34] J. T. Schrauwen, A. Coenen, A. Kurata, *et al.*, "Functional and anatomical measures for outflow boundary conditions in atherosclerotic coronary bifurcations," *Journal of Biomechanics*, vol. 49, no. 11, pp. 2127–2134, Jul. 2016.
- [35] A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in pytorch," 2017.
- [36] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, Mar. 1975.
- [37] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, Sep. 1976.
- [38] I. T. Jolliffe, *Principal Component Analysis*. Springer New York, 1986.
- [39] K. Crane, C. Weischedel, and M. Wardetzky, "The heat method for distance computation," *Commun. ACM*, vol. 60, no. 11, pp. 90–99, Oct. 2017, ISSN: 0001-0782.
- [40] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [41] M. Desbrun, M. Meyer, and P. Alliez, *Intrinsic parameterizations of surface meshes*, 2002.
- [42] M. S. Floater, K. Hormann, and G. Kós, "A general construction of barycentric coordinates over convex polygons," *Advances in Computational Mathematics*, vol. 24, no. 1–4, pp. 311–331, Jan. 2006.
- [43] M. Matyka, Z. Koza, and Ł. Miroslaw, "Wall orientation and shear stress in the lattice boltzmann model," *Computers & Fluids*, vol. 73, pp. 115–123, Mar. 2013.
- [44] S. Ahmad and G. Tesauro, "Scaling and generalization in neural networks: A case study," vol. 1, Jan. 1988, pp. 160–168.
- [45] X. Zhang, H. Li, and Z. Cheng, "Curvature estimation of 3d point cloud surfaces through the fitting of normal section curvatures," *Proceedings of AsiaGraph 2008*, pp. 72–79, Jan. 2008.
- [46] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," Jul. 27, 2016. arXiv: <http://arxiv.org/abs/1607.08022v3> [cs.CV].
- [47] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, Oct. 2016.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," Mar. 16, 2016. arXiv: <http://arxiv.org/abs/1603.05027v3> [cs.CV].
- [49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 22, 2014. arXiv: <http://arxiv.org/abs/1412.6980v9> [cs.LG].
- [50] L. N. Smith, "Cyclical learning rates for training neural networks," Jun. 3, 2015. arXiv: <http://arxiv.org/abs/1506.01186v6> [cs.CV].
- [51] S. M. Ross, "Linear regression," in *Introductory Statistics*, Elsevier, 2010, pp. 537–604.
- [52] D. W. Boyd, "Stochastic analysis," in *Systems Analysis and Modeling*, Elsevier, 2001, pp. 211–227.
- [53] D. Nettleton, "Selection of variables and factor derivation," in *Commercial Data Mining*, Elsevier, 2014, pp. 79–104.