

Automatic Coronary Artery Lumen Border Detection Using Deep Convolutional Neural Networks

Chris Mamon, Vikas Thondapu, Yoong Ong, Peter Barlis, Eric K.W. Poon, Kotagiri Ramamohanarao
and Andrew Ooi

Abstract—In this paper we propose a novel method for delineating the border of the main coronary vessel within Intravascular Optical Coherence Tomography (OCT) images using Convolutional Neural Networks (CNN). Manual segmentation of the main coronary vessel within OCT images is labor intensive and requires, not only, the segmentation of the lumen border but also the detection of side branches. Side branches must be detected and then excluded from the delineated borders as they can influence decision making during percutaneous coronary intervention. Our proposed CNN method incorporates information from the polar-format image stack of an OCT in order to automate the correction of the lumen detected from frames containing side branches. The method has been benchmarked against gold-standard segmentation achieving a mean absolute percentage error of the area of 2.552%, a 0.982 Dice Coefficient and a Hausdorff distance of 0.106 mm. The total time taken to process a single image was estimated at 93 millisecond (ms) on a desktop with an Intel Core i9 7900X @ 3.30GHz CPU with a NVIDIA GeForce GTX 1080 GPU with 64GB ram

Index Terms- Machine Learning, Optical Coherence Tomography, Segmentation

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

E. K.W. Poon is with the Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, Victoria, Australia (e-mail: epoon@unimelb.edu.au)

V. Thondapu is with the Department of Medicine, Faculty of Medicine, Dentistry & Health Sciences, Melbourne Medical School, University of Melbourne, Parkville, 3010 Victoria, Australia, and also with the Department of Medicine, Faculty of Medicine, Dentistry & Health Sciences, Melbourne Medical School, University of Melbourne, Parkville, Victoria, Australia, and also with the Massachusetts General Hospital, Harvard Medical School Cardiovascular Imaging Research Center Boston, Massachusetts, USA (e-mail: vthondapu@mgh.harvard.edu)

Y. Ong is with the Department of Mechanical Engineering, Melbourne School of Engineering, University of Melbourne, Parkville, Victoria, Australia (e-mail: yoongjian1@gmail.com)

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

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

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

I. INTRODUCTION

ISCHEMIC heart disease (IHD) arises from the blockage of one or many arteries which supply the heart with blood and is the single leading cause of death globally (causing approximately 8.9 million deaths in 2016) [2, 3]. Treatment of IHD can be guided by various imaging modalities, the traditional method being coronary angiography [4]. Nevertheless, the method is incapable of depicting the vessel wall making it suboptimal for the assessment of stent implantation or plaque characteristics. One such method which overcomes these limitations, as well as provides accurate lumen dimensions, is intravascular Optical Coherence Tomography (OCT) [5-7]. During the procedure, an OCT catheter is inserted intravascularly and connected to a console. The console automatically pulls back a probe situated inside the OCT shield which obtains high resolution image cross-sections (10-20 μm) of the vessel using near-infrared light, through interferometry, generating a stack of images [1]. Analysis of these images often involves delineation of the lumen (inside space of arterial vessel) and is a key step towards the assessment of IHD and planning of percutaneous coronary intervention (PCI) [8]. However, the pullback can last for several centimeters with a common pullback speed of 10-20 mm/s producing hundreds of images [9]. Consequently, manual delineation of the lumen from OCT images is labor intensive, taking experts between 4 to 7 hours to fully segment a single pullback manually. An algorithmic approach benefits the process two fold as the burden on clinicians as well as the variability between observers are reduced. [10, 11].

Automating the delineation process is often challenging due to the presence of blood that can reduce the clarity of OCT images, a guidewire shadow which partially occludes the boundary [11-14] and side branch ostia that affect main vessel dimensions [15]. Because of the shortfalls, methods which reliably delineate the main vessel lumen border, from the side branches and correct the guidewire shadow is desirable.

Machine learning methods provide a robust approach to tackling lumen segmentation. Machine learning with deep learning techniques have become widely popular for use in computer vision tasks. A specific form of deep learning, the convolution neural network (CNN), represents state-of-the-art performance regarding image-classification and semantic segmentation tasks [16]. As such, works by Miyagawa et al. [17] and Yong et al. [18] show state-of-the-art performance for

lumen segmentation, respectively, using direct acyclic graph (DAG) and linear-regression convolutional neural networks. Yet, despite promising performance demonstrated by Yong et al. [18], errors are often found at side branch ostia where the side branch angles span $> 90^\circ$. To eventually arrive at an approach to fully automatically delineate the lumen, it is important to address the above concerns.

There have been several methods which outline the detection and correction of frames with side branches. Takimura et al. [12] finds a frontier circle around the side branch region using a distance transform. Wang et al. [19] locates a side branch from a 2D matrix containing the distance for all angles along the IPS. Joseph et al. [20] created a log-Maximum Intensity Projection (MIP) of the image pullback stack (IPS) and found the side branches from intensity thresholding. Cao et al. [21] locates the side branch ostium points using a feature of curvature. After locating the side branches the most common method to correct them is to apply a circle approximation to the vessel border. However, by closing the side branches as a circle the natural shape of the artery is not maintained. Thus, research efforts are continuously made to refine and optimize methods which correct the borders while maintaining the overall shape of the vessel.

Another common issue to tackle is errors in the border caused by the guidewire. Correction of the guidewire has been performed using both dynamic programming (DP) and thresholding of either an MIP or 2D accumulated intensity map (2D-AIM) of the OCT IPS [20, 22, 23]. To generate the 2D-AIM the pixels along each A-Line of a polar format cross-section are added together. The resulting columns are then mapped to a rectangle. By detecting the guidewire within such a stack its azimuthal co-ordinates along each border can be located and corrected. Nevertheless, methods involving thresholding can miss some regions whilst DP may overestimate the guidewire size if a side branch or stent overlaps it. Due to the limitations with the described methods, border correction remains an open problem to be tackled.

In this paper we contribute a novel method to accurately delineate the main vessel lumen borders. We use several pre and post processing methods to improve the quality of segmentation by taking into consideration morphology of the guided wire as well as the locations of the side branches. In the proposed method we use CNN for segmentation of intravascular OCT images as well as for both the guidewire and side branches from the 2D-AIM of the IPS. The resulting information about the side branches and guidewire can then be used to correct the borders detected by the CNN.

The paper is organized as follows: details on the CNN materials and method are described in Section II; Section III validates the CNN architecture; Section IV presents the results and finally, the discussion and conclusion are drawn in section V.

II. MATERIALS & METHOD

The proposed method can be divided into three stages: segmentation of guidewire, side branch and lumen borders, post-processing and finally border correction. Further details are described below. The CNN was implemented in a Python environment using Tensorflow v1.12 and run on a desktop with

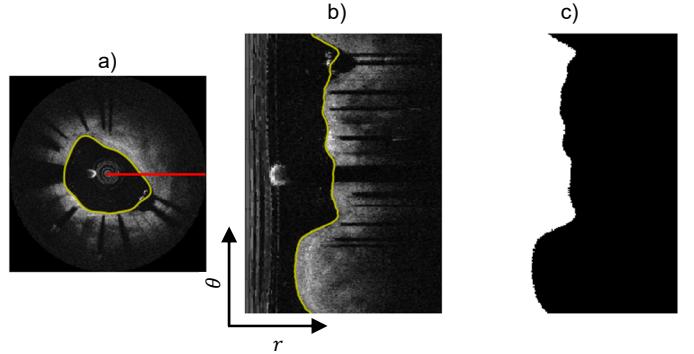


Figure 1: (a) is a polar format image slice from (b). Its accumulated pixel intensities are shown in red in (b). (c) ground truth boundaries of guidewire and side branches colored in yellow, (d) guidewire (e) side branch binary segmentations.

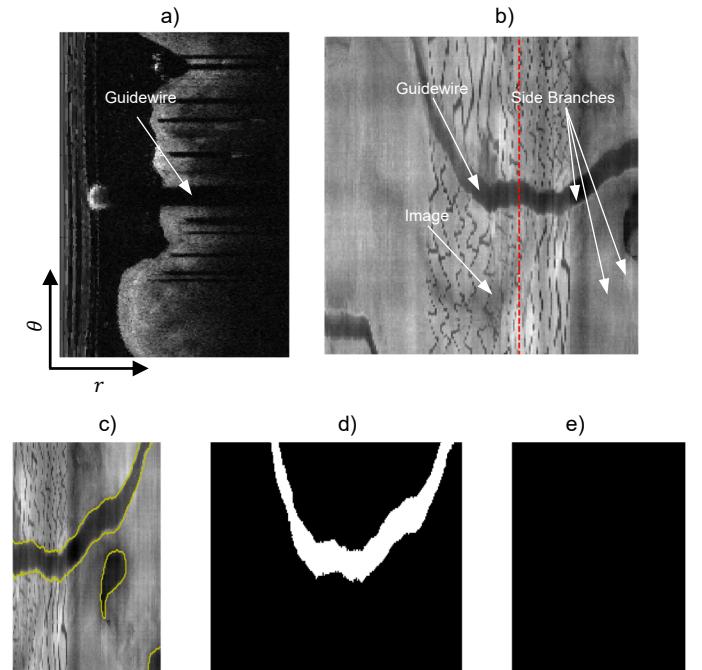


Figure 2: (a) is a polar format image slice from (b). Its accumulated pixel intensities are shown in red in (b). (c) ground truth boundaries of guidewire and side branches colored in yellow, (d) guidewire (e) side branch binary segmentations.

an Intel Core i9 7900X @ 3.30GHz CPU with a NVIDIA GeForce GTX 1080 GPU with 64GB RAM.

A. OCT Data Acquisition and Preparation for Training and Testing

In total, 14 OCT pullbacks from 8 patients were retrospectively selected from a previous multicenter clinical trial (NCT01776567), totaling 3528 images. Each cross-sectional image of the pullbacks was originally 1024×1024 pixels in size, however, they were down sampled to 256×256 to reduce training and processing times. For each image the raw intensities were normalized linearly by the mean and standard deviation. Gold standard lumen border delineation were generated by an experienced clinician outlining the borders frame-by-frame using QIvus (version 3.0, Medis Medical Imaging, Leiden, the Netherlands) according to the standard clinical guidelines [1]. Regions masked by the guidewire shadow and side branches were manually drawn to best

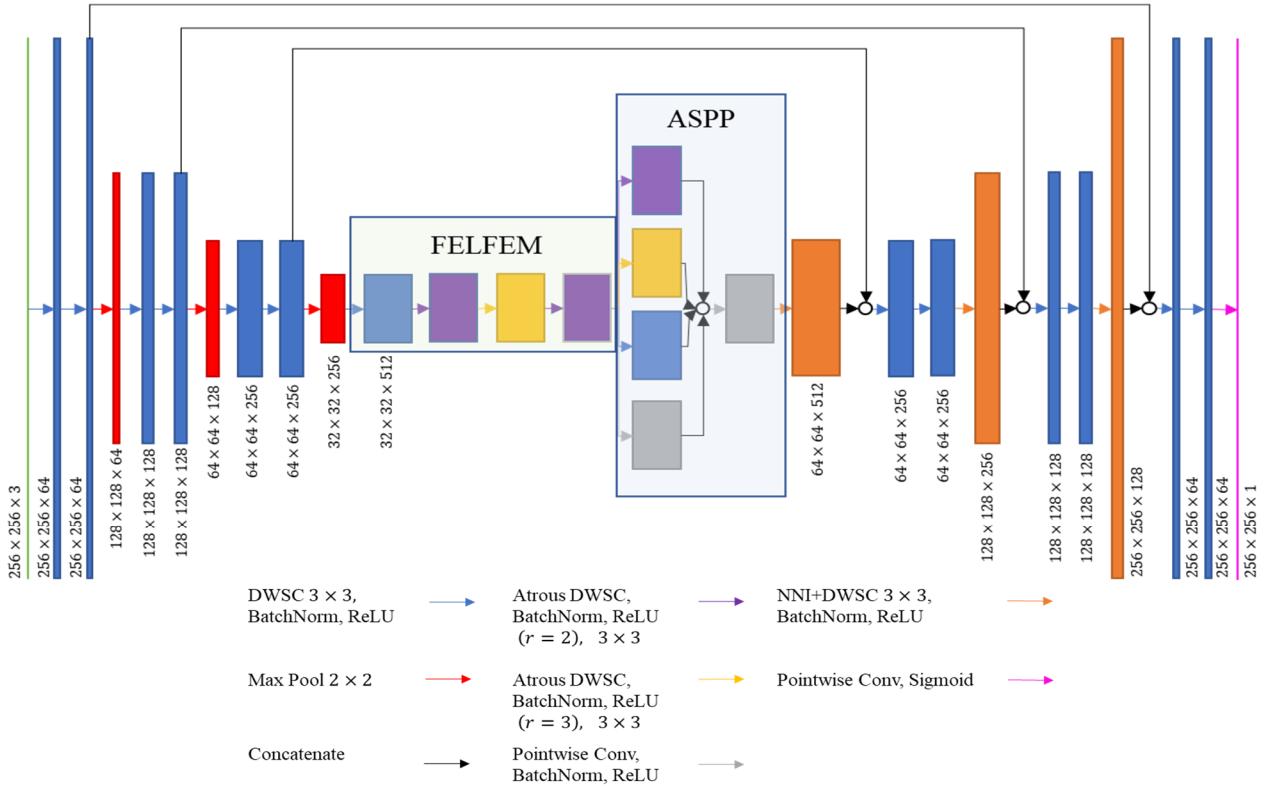


Figure 3: Shows U-net architecture for final Model E. Each box corresponds to a multi-channel feature map with dimensions width \times height \times channels. The color of each arrow and box denote different operations whilst the small white circles correspond to the concatenation of multiple feature map.

estimate the underlying lumen border. These contours were then used to generate binary segmentation maps in polar coordinates of the lumen for the training procedure, Figure 1. From the original polar space IPS a 2D-AIM is produced by accumulating pixel intensities over individual A-lines [23], Figure 2a) and b). Figure 2d)-e) show binary segmentations of side branches and guidewires from the boundaries shown in Figure 2a).

Data is divided at the patient level into a 70% and 30 % train and test-split. From the training data 3 patients are randomly chosen to perform 3-fold cross validation to validate the choice in neural network architecture as well as determine the filter sizes used during the post-processing stage.

B. Convolutional Neural Network Architecture and Training

Input image pixels are normalized linearly between 0 and 1 before being fed into the neural network. The neural network consists of an encoder-decoder [24-26] with long-range skip connections [27], taking on the structure of a U-Net [28, 29]. Depthwise Separable Convolutions (DWSC) are used exclusively as they have been used in prior work to reduce network complexity, and allow for future usage in mobile devices [30-32]. The activation function following most convolution operations is the rectified linear unit (ReLU). We have also replaced the transposed convolutions with Nearest Neighbor Upsampling followed by DWSC (NNU+DWSC) to reduce checkerboarding artifacts [33]. The middle layers are Front End and Local Feature Extraction Modules (FELFEM) with $r = (1,2,3,2)$ [34] leading into an Atrous Spatial Pyramid Pooling ASPP layer with rates $r = (1, 2, 3)$ [35]. The FELFEM and ASPP layers replace the usual deeper max pooling layers

contained in the U-net [36]. This has been performed to reduce signal decimation and increase the receptive field [37]. FELFEM layers replace the traditional increased dilation factor used by the original architecture developed by Chen et al. [35]. FELFEM can reduce the effect of “jagged patterns” which occur when using an increasing dilation factor exclusively [34, 38]. For the main body of convolution operations batch normalization is applied to accelerate training and allow the use of larger learning rates [39]. The network is trained to perform pixelwise class labelling of the lumen (lumen 1 not lumen 0, i.e. binary semantic segmentation). The architecture used can be observed in Figure 3.

For each convolution operation images are padded with cyclic boundary conditions in the azimuthal direction, and 0 padding in the radial direction. This ensures that the CNN can properly infer the correct boundary if the guidewire or side branches is located directly on the edge of the image.

C. Loss Function and Training Protocol

As we have three features (lumen border, guidewire, side branches) to segment from two images sets, three sets of weights are generated for the designed CNN. The three sets of weights are trained using the same loss function, but, are trained using two different optimization methods which will be described. This is because it was found that for the 2D-AIM data, applying Stochastic Gradient Descent with Warm Restarts (SGDWR) did not improve over stock Adam-Optimization.

The loss function used was a combination of the binary cross entropy (BCE) and the log of the Dice coefficient (DICE). The metrics BCE and DICE are combined in a fashion described by Iglobikov et al. [26]. By doing so the probabilities of the correct

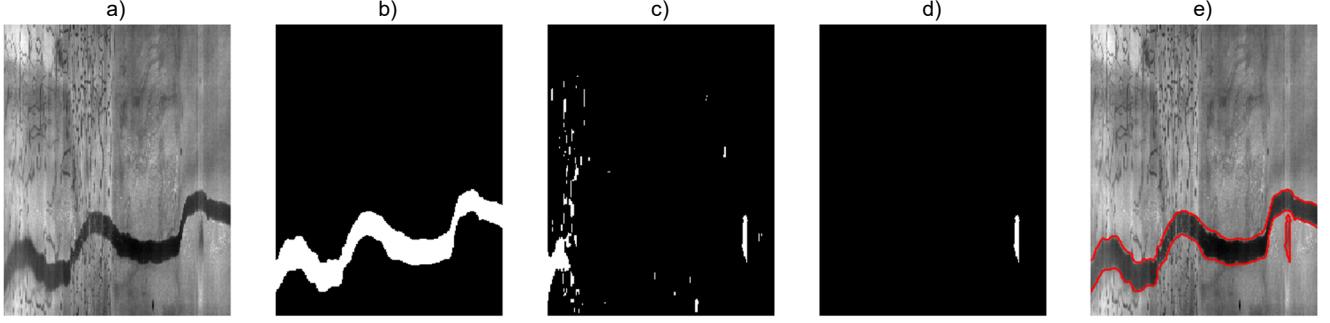


Figure 4: (a) input image, (b) detected guidewire, (c) is the bifurcations detected by the CNN in the 2D-Aim. In (d) false positives are removed using the guidewire and morphological operations. (e) shows the resulting guidewire and bifurcation segmentations by the CNN.

pixel as well as the intersection being predicted is maximized during training.

DICE is the most common metric used in validating medical imaging segmentations [40]. It is used to gauge the similarity between two samples and is defined as,

$$DICE = 2 \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i} \quad (1)$$

where y_i is the binary value, i.e. label, of the pixel i and \hat{y}_i is the predicted probability of i .

The BCE function is a common loss function used for training CNN. It has a proven track record for accelerating the convergence of the back-propagation algorithm and provides good results. It is defined as,

$$BCE = -\frac{1}{n} \sum_{i=1}^n (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)) \quad (2)$$

the combined equation BCE+DICE becomes,

$$LOSS = BCE - \alpha \ln(DICE) \quad (3)$$

where α is a tunable hyper parameter set to 1.

Segmentation of the lumen border was trained with SGDR: Stochastic Gradient Descent with Warm Restart (SGDWR) [41]. During this process training is restarted after T_n epochs are performed, where n is the index of the run. Within the run n the learning rate is decayed with cosine annealing for each batch as follows,

$$\eta_t = \eta_{min}^n + \frac{1}{2}(\eta_{max}^n - \eta_{min}^n) \left(1 + \cos \left(\frac{T_{cur}}{T_n} \pi \right) \right) \quad (4)$$

where η_{max}^n and η_{min}^n are the learning initial and final learning rates of the run, T_{cur} is the number of epochs performed since the last restart. T_n starts with an initially small value and is increased by T_{mult} after each restart such that $T_n = T_{n-1} T_{mult}$. η_{max}^n has an initial learning rate of η_{max}^0 and is scaled by a value m at every restart such that $\eta_{max}^n = \eta_{max}^{n-1} m$.

The values chosen for $\eta_{min}^0 = 0$, $\eta_{max}^0 = 0.01$ and $m = 0.5$. While testing different architectures the values for T_n chosen are $(T_0, T_1, T_2) = (1, 2, 4)$ with $T_{mult} = 2$. When training over the complete dataset $T_n = (2, 4, 8, 16)$.

For the 2D-AIM dataset the pre-trained weights from the OCT dataset is used so help speed up training [26, 42]. An Adam-Optimizer is used with an initial learning rate of 0.01.

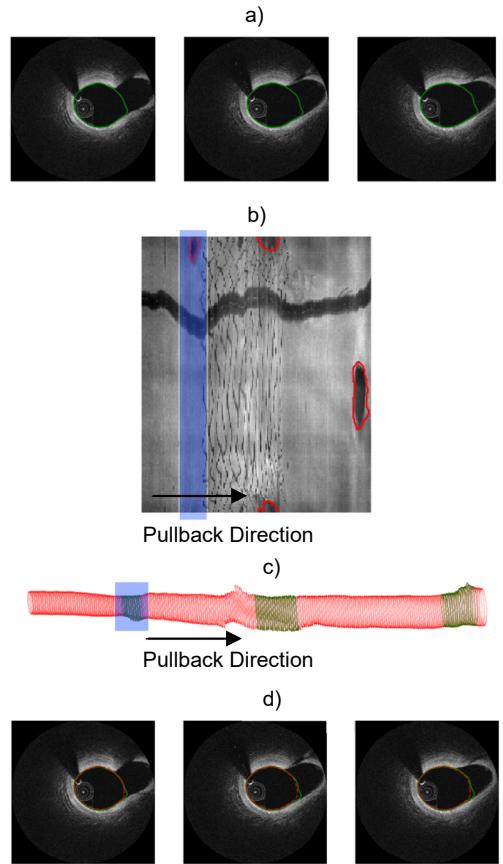


Figure 5: (a) contours to be corrected, located at a side branch. (b) 2D-AIM with detected side branches and guidewire highlighted in red whilst region containing frames in (a) has been highlighted in blue. (c) the contour stack where all detected bifurcations are highlighted in green and the region containing the images in (a) is highlighted in blue. (c) the initial contours are colored in green whilst the filtered contours are highlighted in red

The training was run continuously for 500 epochs where convergence was observed, i.e. the loss ceased to improve.

Images are augmented during training time with random rotations, random cropping, pixel dropout, elastic deformations and color change to increase the number and variations of training examples, which has been shown to improve training outcomes [36, 43]. A batch size of 4 was chosen to fit within the size of the GPU memory.

The trained neural networks were then used to predict the lumen border and locate the guidewire and side branches for the testing sets.

D. Post-Processing & Border Correction

After the OCT and 2D-AIM have been segmented with the CNN, false positives are removed from the binary segmentation maps with morphological operations. Morphological opening and closing operations were used with a disk-shaped structuring element set to 5. This was chosen in accordance with Pociask et al. [44] who determined the value experimentally. This helps to minimize any artifacts during the binary segmentation.

The lumen borders of the binary segmentation maps are detected through the use of canny edge detection [45]. The resulting borders are interpolated in polar co-ordinates through the use of b-spline curves [46]. The detected borders may not be smooth as the border location is based on the integer pixel location, thus, a Savitzky–Golay [47] sliding polynomial filter with window width 21 and polynomial order 3 is applied in a similar fashion to Pociask et al [44].

The segmentation of the side branches is not as accurate as the segmentation of the guidewire and false positives appear especially around the guidewire. To remove these false positives the difference between the binary segmentation of the side branches and guidewire is taken and the resulting values are clipped to between 0 and 1. Morphological operations are then applied to remove any final small speckles, Figure 4c), to ensure only borders with large side branches are corrected as shown in Figure 4e).

All the contours are then stacked. From Figure 5 we can observe that if we align the contour stack with the 2D-AIM each pixel along the pullback direction matches 1-to-1 to an OCT image. Thus, the 2D-AIM (Figure 5b) is used to locate the contours containing side branches by identifying the columns along the pullback direction containing pixels with a value of 1. Once these columns are identified they can be grouped. The groups of contours can be seen in Figure 5c). Each side branch group is then isolated and a 2D Savitzky–Golay filter with a window size of (21, 21) and a polynomial order of 3 is applied to each with a constant boundary condition in the pullback direction. The results of the filtering is shown in Figure 5d).

After correcting the images with side branches and smoothing the borders, there may still be artefacts due to the guidewire. In the final stage of post-processing the azimuthal location of the guidewire is found for each image. The region of each border is removed and interpolated in a similar fashion to methods outlined in the literature, however in our case, using the binary segmentation from the CNN [11, 19, 22].

III. ARCHITECTURE VALIDATION

We first compare the effect of various architecture changes on the performance of the neural network. The new model has been compared with the original U-Net implementation [36]. We split the data at the patient level and three patients have been chosen to assess the effects of changing the architecture. Each patient contained between 200 to 300 images. Three-fold cross validation is performed, and the mean values are reported. Border correction was skipped in this stage as the post-processing portion of the algorithm heavily relies on the quality of initial segmentations by the CNN. The ICE and Hausdorff Distance (HD) are used to assess the segmentations. The HD is a distance-based metric which is commonly used to measure the similarity between two sets of points [48]. It provides extra

information over DICE by considering the spatial position of the points and not just the overlap. This makes it a very useful method for quantifying the variation between contours of both segmentation methods. To stay consistent with Taha and Handbury [40] the HD between two finite set points A and B is defined as,

$$HD(A, B) = \max(h(A, B), h(B, A)) \quad (5)$$

where $h(A, B)$ is the directed Hausdorff distance,

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (6)$$

taking $\|a - b\|$ as the Euclidean distance.

A. Comparison of Neural Network Performance With Various Modules

Model	DICE	HD (mm)	Parameters 10^6
Baseline	0.962	0.182	
Transposed Conv	± 0.059	± 0.229	31.04
Baseline	0.975	0.155	
NNU+DWSC	± 0.052	± 0.209	28.98

Table 1: Comparison of performance of U-net with regular convolutions using either transposed convolutions or NNU+DWSC. Mean \pm standard deviation of the DICE (higher better) and Hausdorff distance (lower better) is shown.

We compare the effect of transposed convolutions vs NNU+DWSC. From Table 1 we observe an improvement for both the DICE and HD. To perform fair comparison of the models, tests on models A, B and C were performed to assess the effect of the number of trainable parameters on the model performance. Models A, B and C are the baseline U-net however the regular convolutions are replaced with DWSCs and the transposed convolutions are replaced by NNU+DWSC. Models D and E shows how the performance of the model changes when ASPP is or is not applied after FELFEM. Table 2 gives architecture details of models A-E.

Model	Largest Stride	Base Filters	FELFEM (1,2,3,2)	ASPP (1,2,3)
A	16	32	-	-
B	16	64	-	-
C	16	96	-	-
D	8	64	Yes	-
E	8	64	Yes	Yes

Table 2: Comparison of model types. Model A, B, C are regular U-nets with DWSC and NNU+DWSC. Models D and E do not have the same final output stride however the middle flow is replaced by FELFEM or FELFEM+ASPP

Model	DICE	HD (mm)	Parameters 10^6
A	0.936 ± 0.102	0.252 ± 0.298	1.07
B	0.951 ± 0.078	0.219 ± 0.288	3.93
C	0.959 ± 0.047	0.180 ± 0.270	8.75
D	0.970 ± 0.043	0.154 ± 0.238	1.53
E	0.976 ± 0.032	0.142 ± 0.207	3.64

Table 3: Comparison of performance of models A-E. The mean \pm standard deviation of the DICE (higher better) and Hausdorff distance (lower better) is shown.

As observed from Table 3, model E provides the best quantitative results when compared with the other models. Thus, we use E for reporting our results in the following section. Nevertheless, we must note that the architecture was tested on a limited training set to save time.

IV. RESULTS

Dataset	Training Time (hr:min:sec)
Lumen Detection	03:21:12
Guidewire and Side Branches	00:32:42

Table 4: Summary of the time taken to train the CNN to detect both the lumen and the side branches.

The CNN was trained on 2444 cross-sectional images and 10 2D-AIM. The accuracy of our proposed CNN lumen segmentation algorithm was validated against gold-standard segmentations of the test data, totaling 1084 cross sectional images and 4 2D-AIM. The accuracy of side branches detection is assessed by analyzing the total number of correctly located frames. The guidewire and overall lumen segmentation algorithm are assessed by comparing point by point distance error and binary overlap. A summary of the time taken for training is shown in Table 4.

The algorithm depends on correctly identifying frames which contain side-branches, thus, we count the number true positive and false positive frames to assess the algorithm. Over the total 1084 frames, 115 frames contained side branches. Side branches in 94 frames were correctly identified, whilst 18 frames were falsely identified as containing side branches. Thus, there were 94 true positive (TP), 18 false positive (FP), 21 false negative (i.e. out of the 115 frames 21 frames are falsely identified to contain no side branches) (FN) and 950 true

negative (TN) identified frames. A summary of the accuracy, sensitivity, positive predictive value (PPV), negative predictive value (NPV) and F1 score are summarized in Table 5.

Side Branch Frame Detection					
Accuracy	Sensitivity	Specificity	PPV	NPV	F1 Score
0.963	0.817	0.981	0.839	0.978	0.828

Table 5: Accuracy, sensitivity, positive predictive value (PPV), negative predictive value (NPV) and F1 score (higher better)

As the exact location of the guidewire is used to correct the contour, accurately calculating the start and end points of the shadow in the azimuthal direction is important. To assess how accurately the CNN matches with manual segmentation the DICE is calculated to compare the area. The difference between the contours outlining the guidewire of the two methods is calculated and compared frame by frame. To convert the difference in pixel location to angle we multiply the results by $360^\circ/256$ pixels to get the difference in degrees. These results are summarized in Table 6. The outcome of applying the Savitzky–Golay filter [47] along the contour stack over the detected frames is shown in Figure 7. Full segmentation of the side branches and guidewire are shown in Figure 6.

Guide-Wire Shadow Detection	
DICE	Angle
0.977 ± 0.005	$0.639 \pm 0.965^\circ$

Table 6: Mean \pm standard deviation of guidewire detection scores, the DICE (higher better) and difference in azimuthal location of guidewire boundary in degrees (lower better).

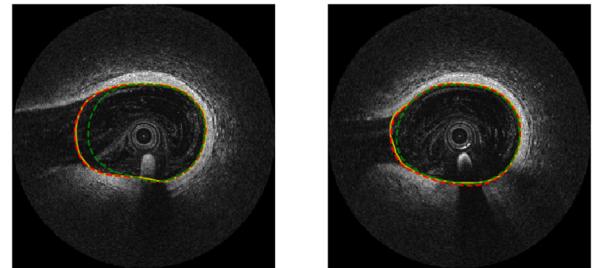
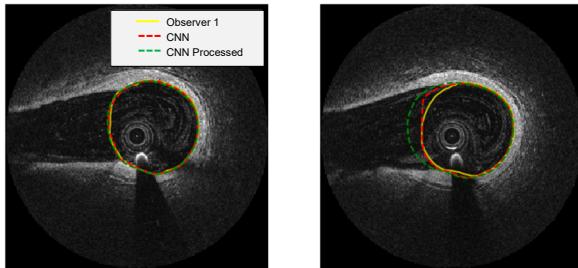


Figure 7: Shows contours at a side branch which have been corrected by the Savitzky–Golay filter. Yellow contour shows the ground truth by observer 1, red shows the original CNN segmentation and green shows the final post processed contours.

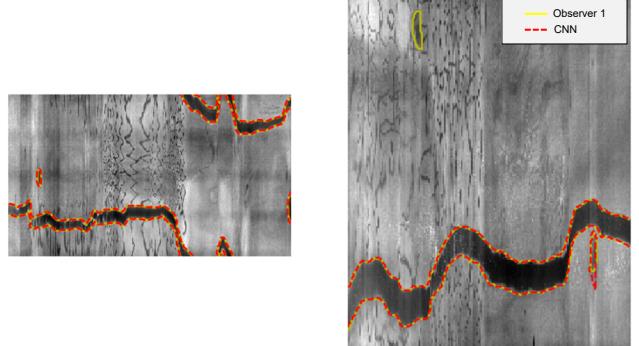
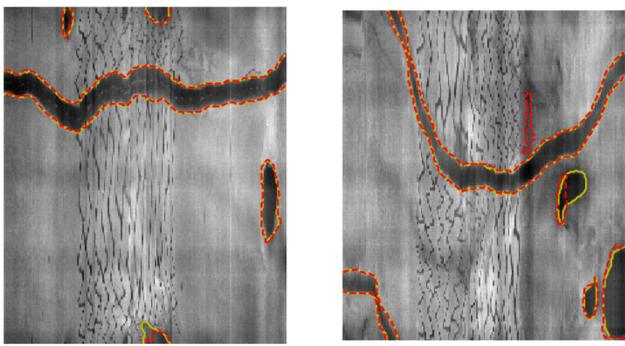


Figure 6: Segmentation of guidewire and side branches. The gold lines show gold standard segmentations whilst the red lines show the results of the CNN

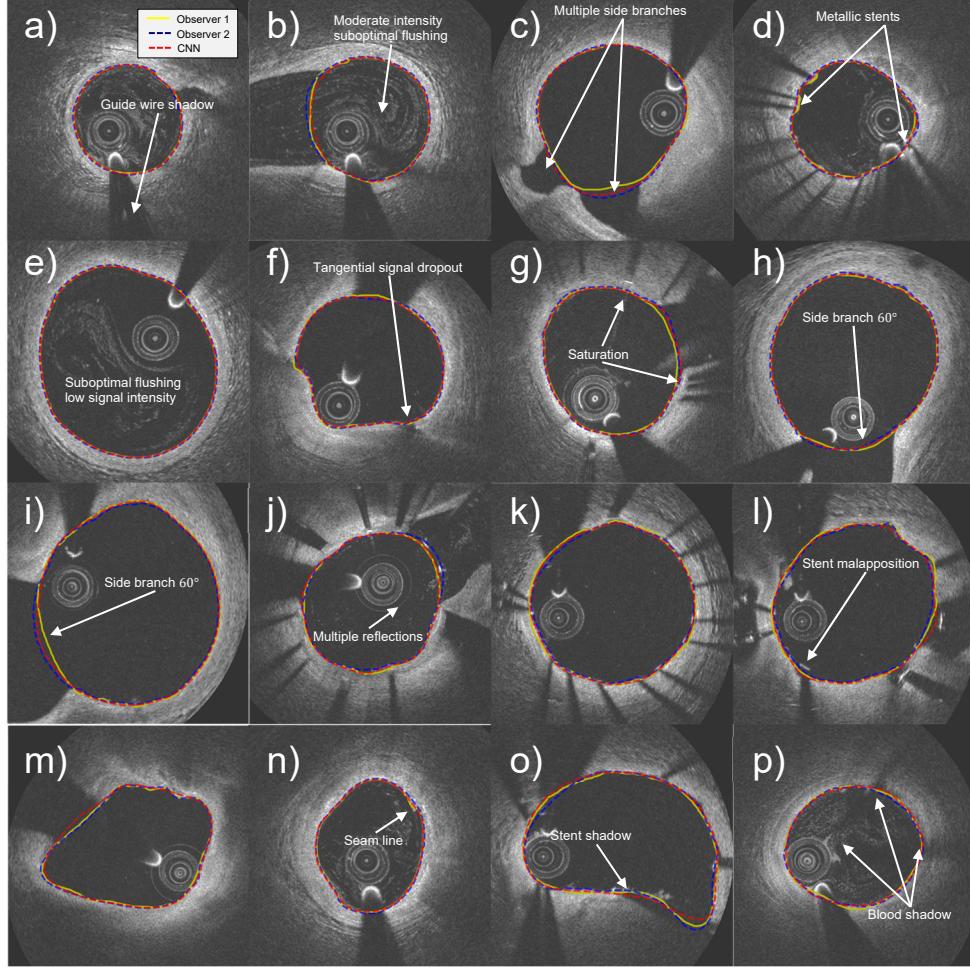


Figure 8: Example results from the test sets with various artefacts, and segmentation challenges as described in the document of consensus [1]. Observer 1 is highlighted in yellow Observer 2 is highlighted in blue and the CNN is highlighted in red.

To compare the point by point difference of the CNN contours to the gold standard (ground truth) contours, each contour is divided into 360 points, equally spaced in the azimuthal direction. The overlap of each image contour is assessed by binarizing the lumen and calculating the DICE. The HD for the predicted contours and the gold standard were computed for all images in polar co-ordinates. Finally, the lumen area is compared by calculating the absolute percentage error (APE) of the area. The resulting values can be observed in Table 7.

Lumen Contour Detection			
	DICE	HD (mm)	APE (%)
All	0.982 ± 0.042	0.106 ± 0.163	2.552 ± 2.631
Side Branch (not corrected)	0.955 ± 0.076	0.234 ± 0.197	6.871 ± 6.72
Side Branch (corrected)	0.963 ± 0.026	0.187 ± 0.147	5.427 ± 5.94

Table 7: OCT Image contour detection metrics. We show mean \pm standard deviation for DICE, APE (higher better) and Hausdorff Distance (lower better) for the set of all contours and subset of contours containing side branches when corrected and when not corrected.

To measure the linear relationship and agreement between the contour detection algorithms Pearson's Correlation Coefficient (Pearson's r) and Bland-Altman analysis was used. All 1084 images were segmented by two independent

observers (whom we have labelled as Observer 1 and Observer 2) and the CNN, examples can be seen in Figure 8. The segmentations were then converted into polar coordinates and 360 points were sampled at identical azimuthal locations. One thousand radial points from the contours are then sampled randomly for comparison. Pearson's r is calculated for Observer 1 vs Observer 2 and Observer 1 vs CNN. Bland-Altman analysis is then performed between observer 1 and the CNN. We analyzed the distribution of differences between the two measures; areas and point by point radial distance of image contours. It was found, by the Kolmogorov Smirnov test [49] that the differences were not normally distributed, $p < 0.001$. Hence we employ non-parametric analysis to generate the Bland-Altman plots [50]. The limits of agreement are set to the 97.5 and 2.5 percentile, showing regions where 95 percent of the difference between the two measures lie for varying means between the observers.

The linear relationship between Observer 1 and Observer 2 was measured for the area and radius using Pearson's r . It was found to be 0.984 for the area and 0.978 for the radius. For

Method	Test Size	Specified Info	Reported DICE
Cheimariotis et al. [51] (1s)	1812 images from 20 pullbacks	No Exclusions	Mean DICE: 0.935 (stented), 0.925 (non-stented)
Gurmeric et al. [52] (5.9 ± 3s)	39 cross sections	Included Neo Intimal Hyperplasia, excluded heavy blood clotting	Mean DICE: 0.971
Yong et al. [18] (40.6 ms)	45 training pullbacks with 13,342 images, 19 test pullbacks with 5686 images	No Exclusions	Median DICE (Interquartile Range): 0.985 (0.979, 0.988)
Wang et al. [19]	106 images selected randomly from 9 pullbacks	Applied to cases with stents, side branches and small amounts of luminal blood.	Mean DICE: 0.97
Maysa et al. [12] (15s)	1328 frames from 9 pullbacks	Excluded stented and substantial luminal blood slices	Mean DICE ± standard deviation: Outside bifurcation regions 0.973 ± 0.027 Inside bifurcation regions 0.905 ± 0.097
Miyagawa et al. [17]	1689 images from 9 pullbacks divided into 5-fold cross validation	No description	Mean DICE ± standard deviation: 0.988 ± 1.26 (data not divided at patient level)
Cao et al. [11]	880 images from 4 patients	Blood artifacts excluded, Bifurcations are ignored	Mean DICE ± standard deviation: 0.981 ± 0.011
Moraes et al. [53] (6s)	290 images	Irregularities and eccentricity of lumen due to thrombus, plaques, branches, several tissue contrasts, and stent implanted	Mean DICE ± standard deviation: 0.978 ± 0.0216
Present study (93 ms)	1084 images from 4 pullbacks	No Exclusions	Mean DICE ± standard deviation: 0.982 ± 0.042 Median DICE (Interquartile Range): 0.987 (0.981, 0.991)

Table 8: Summary of methods which have reported DICE. The results of our works have been shown in the last row. The time taken to segment a frame has been included next to the author when specified. We have also included the median DICE of the present study for comparison with papers providing the median.

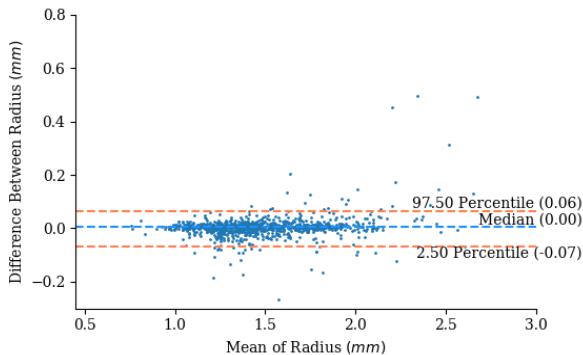
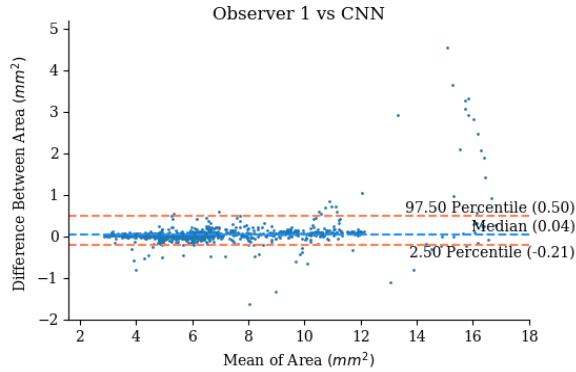


Figure 9: Bland-Altman plot analysis of luminal area for all segmented areas between observer 1 and the CNN as well as the radial distance for 1000 randomly sampled points.

Observer 1 and CNN Pearson's r was calculated as 0.991 and 0.989 for area and radius. From the Bland-Altman analysis shown in Figure 9, the estimated median difference of observer

1 vs CNN for area and radial distance is -0.04 mm^2 and 0.00 mm respectively, while the 95 percent range is $(-0.21, 0.50) \text{ mm}^2$ and $(-0.06, 0.08) \text{ mm}$. As there is not a reference dataset for comparison, results of recent methods have been summarized in Table 8.

The total time taken per step to complete a segmentation from a single pullback has been summarized in Table 9. The time taken for a pullback with 300 images is estimated to be approximately 28 seconds. This time does not include overheads such as loading modules in python or saving the resulting data to disk. Thus, we estimate that the algorithm will take 93 ms to detect the border per image on a desktop with an Intel Core i9 7900X @ 3.30GHz CPU, NVIDIA GeForce GTX 1080 GPU with 64GB ram.

Time Taken (ms)	
Guidewire	88.682 ± 11.322
Side Branch	89.732 ± 11.473
Lumen Contour (per image)	81.501 ± 7.713
Post-Processing (per image)	10.804 ± 4.514

Table 9: Summary of time required for each step in the reconstruction. Times for guidewire and side branch shows the mean ± standard deviation of time taken to segment each pullback. The Lumen Contour and Post-Processing times show the mean ± standard deviation of time taken to segment a single frame in a pullback.

V. DISCUSSION AND CONCLUSION

Lumen analysis is a key step towards the assessment of IHD and planning of PCI [8]. In this work, we presented a novel automatic method for segmenting the lumen from OCT images. The method combines machine learning and post-processing to tackle problems, experienced by Yong et al. [18], of errors within contours at side branch regions spanning an angle $> 90^\circ$.

degrees. Initially to tackle this issue we designed a novel neural network architecture using principles inspired by Hamaguchi et al. [34], Chen et al. [35] and Ronneberger et al. [36]. The architecture implemented shows improvements over the classic U-Net (0.976 vs 0.962 DICE) [36] with a similar level of complexity as that of mobile nets [30] (our model E has 3.64 million parameters, while 1.0 MobileNet-224 has 4.2 million parameters). Using this architecture, we segmented the lumen contour from OCT images and applied morphological operations and targeted filtering to the contours. This targeting was performed by detecting bifurcations from a 2D-AIM using a CNN trained via transfer learning. Our results indicate that the CNN combined with targeted filtering can improve the segmentation performance of a CNN model (DICE 0.955 to 0.963 when applying filtering at side branches). The overall algorithm showed good agreement with that of trained observers (Pearson's r for Observer 1 vs Observer 2; 0.984, 0.978 for area and radius, and Observer 1 vs CNN; 0.991, 0.989 for area and radius).

Our results are comparable with works from Miyagawa et al. [17] and Yong et al. [18], showing a slight improvement over Yong et al. When analyzing images at side branches our results show improvements over works by Maysa et al. [12]. It must, however, be noted that the dataset used was significantly smaller than Yong et al [18]. For a better comparison of works a comprehensive universal dataset is required.

Many non-CNN algorithms can take one or more seconds to segment a single image [12, 44, 51-54]. Our algorithm's segmentation time per frame was estimated at 93 ms on a high-performance desktop using a GPU. The time taken for several algorithms to complete a single frame has been summarized in Table 8. By comparison, our algorithm is significantly faster than most non-CNN algorithms. This can allow for vessel lumen assessment in an intraoperative time frame. The main drawback for using a CNN is that the training time is significant, taking 3 hours to train the neural network on 2444 images. The benefit of going through all this effort is that once trained the CNN's weights can be reused and fine-tuned as more data becomes available, allowing our algorithm to improve in future as more data is gathered.

Automatic side branch detection helped with improving contour detection, however, it was shown to perform poorly on images containing blood artifacts. These frames show up as a long dark patch along the 2D-AIM, Figure 6, causing false positives. To potentially improve side branch detection cross-sectional information will be used in future works to identify false positives. One such method which can be used is works by Miyagawa et al [42].

We note the images used did not contain very heavy blood clotting and alternative stent types. The method was also performed over baseline and follow up cases. Because of this, the resulting segmentation may be better than expected. Nevertheless, this indicates that the network can be fine-tuned if segmentation is required on a follow-up case.

A limitation of our method is that the effectiveness of the filter is dependent on the continuity of the images. Thus, the effectiveness of this method will be linked to the framerate of the camera. Lower frame rates will produce less continuity between images, causing higher values of error. Secondly, minor side branches and the lip of major side branches are not

detectable. Thus, any errors caused by them cannot currently be corrected. Finally, the size of the structuring element of the morphological operations used in post-processing may limit the types of cases the algorithm is applicable with. The described issues may potentially be mitigated by using a machine learning based smoothing algorithm. This network would maximize a function of the surface's smoothness and minimize the difference of the smoothed contours to the original contour.

In this paper we described a novel automatic method to segment the guidewire, detect side branches, and accurately delineate the main vessel lumen borders. Our results demonstrate that the developed algorithm is fast and comparable to two trained observers. Future works will apply the algorithm on a larger dataset and develop a machine learning based smoothing method.

VI. BIBLIOGRAPHY

1. Tearney, G.J., et al., *Consensus standards for acquisition, measurement, and reporting of intravascular optical coherence tomography studies: a report from the International Working Group for Intravascular Optical Coherence Tomography Standardization and Validation*. J. Am. Coll. Cardiol., 2012. **59**(12): p. 1058-1072.
2. Mendis, S. and W.H. Organization, *Global status report on noncommunicable diseases 2014 / lead author, Shanthi Mendis*. 2014: World Health Organization Geneva, Switzerland. xvii, 280 pages .-xvii, 280 pages .
3. Organization, W.H. *Cardiovascular diseases (CVDs)*. 2019; https://www.who.int/cardiovascular_diseases/en/.
4. Raber, L., et al., *Clinical use of intracoronary imaging. Part I: guidance and optimization of coronary interventions. An expert consensus document of the European Association of Percutaneous Cardiovascular Interventions*. EuroIntervention, 2018. **14**(6): p. 656-677.
5. Barlis, P., et al., *Assessment of culprit and remote coronary narrowing using optical coherence tomography with long-term outcomes*. Am J Cardiol, 2008. **102**(4): p. 391-5.
6. Karanasos, A., et al., *Optical Coherence Tomography: Potential Clinical Applications*. Current Cardiovascular Imaging Reports, 2012. **5**(4): p. 206-220.
7. Miyamoto, Y., et al., *Plaque Characteristics of Thin-Cap Fibroatheroma Evaluated by OCT and IVUS*. JACC: Cardiovascular Imaging, 2011. **4**(6): p. 638-646.
8. Adriaenssens, T., et al., *Automated detection and quantification of clusters of malapposed and uncovered intracoronary stent struts assessed with optical coherence tomography*. Int J Cardiovasc Imaging, 2014. **30**(5): p. 839-48.
9. Roleder, T., et al., *The basics of intravascular optical coherence tomography*. Postepy Kardiol Interwencyjnej, 2015. **11**(2): p. 74-83.
10. Wang, Z., et al., *Image Processing in Intravascular OCT*, W. Drexler and J.G. Fujimoto, Editors. 2015, Springer International Publishing. p. 477-504.
11. Cao, Y., et al., *Automatic Lumen Segmentation in Intravascular Optical Coherence Tomography Images Using Level Set*. Comput Math Methods Med, 2017. **2017**: p. 4710305-4710305.
12. Macedo, M.M.G.d., et al., *A robust fully automatic lumen segmentation method for in vivo intracoronary optical coherence tomography*. Research on Biomedical Engineering, 2016. **32**: p. 35-43.
13. Chiastra, C., et al., *Patient-Specific Modeling of Stented Coronary Arteries Reconstructed from Optical Coherence Tomography: Towards a Widespread Clinical Use of Fluid Dynamics Analyses*. J Cardiovasc Transl Res, 2018. **11**(2): p. 156-172.
14. Kauffmann, C., P. Motreff, and L. Sarry, *In vivo supervised analysis of stent reendothelialization from optical coherence tomography*. IEEE Trans Med Imaging, 2010. **29**(3): p. 807-818.
15. Paoletti, G., et al., *Reproducibility of serial optical coherence tomography measurements for lumen area and plaque components in humans (The CLI-VAR [Centro per la Lotta Contro l'Infarto-*

- variability] II study). Int J Cardiovasc Imaging, 2016. **32**(3): p. 381-7.
16. Garcia-Garcia, A., et al., *A Review on Deep Learning Techniques Applied to Semantic Segmentation*. CoRR, 2017. **abs/1704.06857**.
17. Miyagawa, M., et al. *Lumen Segmentation in Optical Coherence Tomography Images using Convolutional Neural Network*. in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE.
18. Yong, Y.L., et al., *Linear-regression convolutional neural network for fully automated coronary lumen segmentation in intravascular optical coherence tomography*. J Biomed Opt, 2017. **22**(12): p. 1-9.
19. Wang, A., et al., *Fully automated side branch detection in intravascular optical coherence tomography pullback runs*. Biomed Opt Express, 2014. **5**(9): p. 3160-73.
20. Joseph, S., A. Adnan, and D. Adlam, *Automatic segmentation of coronary morphology using transmittance-based lumen intensity-enhanced intravascular optical coherence tomography images and applying a localized level-set-based active contour method*. J Med Imaging (Bellingham), 2016. **3**(4): p. 044001.
21. Cao, Y., et al., *Automatic Side Branch Ostium Detection and Main Vascular Segmentation in Intravascular Optical Coherence Tomography Images*. IEEE Journal of Biomedical and Health Informatics, 2018. **22**(5): p. 1531-1539.
22. Wang, Z., et al. *Automatic segmentation of intravascular optical coherence tomography images for facilitating quantitative diagnosis of atherosclerosis*. in *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XV*. SPIE.
23. Ughi, G.J., et al., *Fully automatic three-dimensional visualization of intravascular optical coherence tomography images: methods and feasibility in vivo*. Biomed Opt Express, 2012. **3**(12): p. 3291-303.
24. Badrinarayanan, V., A. Kendall, and R. Cipolla, *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. 2015.
25. Chen, L.-C., et al., *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018.
26. Iglovikov, V. and A. Shvets, *TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation*. 2018.
27. Drozdzal, M., et al., *The Importance of Skip Connections in Biomedical Image Segmentation*. 2016.
28. Pandey, R.K., A. Vasav, and A.G. Ramakrishnan, *Segmentation of Liver Lesions with Reduced Complexity Deep Models*. 2018.
29. Han, X., *Automatic Liver Lesion Segmentation Using A Deep Convolutional Neural Network Method*. 2017.
30. Howard, A.G., et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017.
31. Chollet, F., *Xception: Deep Learning with Depthwise Separable Convolutions*. 2016.
32. Zhang, X., et al., *ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices*. 2017.
33. Odena, A., V. Dumoulin, and C. Olah, *Deconvolution and Checkerboard Artifacts*. Distill, 2016. **1**(10).
34. Hamaguchi, R., et al., *Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery*. 2017.
35. Chen, L.-C., et al., *Rethinking Atrous Convolution for Semantic Image Segmentation*. 2017.
36. Ronneberger, O., P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015.
37. Le, H. and A. Borji, *What are the Receptive, Effective Receptive, and Projective Fields of Neurons in Convolutional Neural Networks?* 2017.
38. Yu, F., V. Koltun, and T. Funkhouser, *Dilated Residual Networks*. 2017.
39. Ioffe, S. and C. Szegedy, *Batch normalization: accelerating deep network training by reducing internal covariate shift*, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. 2015, JMLR.org: Lille, France. p. 448-456.
40. Taha, A.A. and A. Hanbury, *Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool*. BMC medical imaging, 2015. **15**: p. 29-29.
41. Loshchilov, I. and F. Hutter, *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2016.
42. Miyagawa, M., et al., *Detecting Vascular Bifurcation in IVOCT Images Using Convolutional Neural Networks With Transfer Learning*. IEEE Access, 2019. **7**: p. 66167-66175.
43. Çiçek, Ö., et al., *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*. 2016.
44. Pociask, E., et al., *Fully Automated Lumen Segmentation Method for Intracoronary Optical Coherence Tomography*. J Healthc Eng, 2018. **2018**: p. 1414076.
45. Cannby, J., *A Computational Approach to Edge Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **PAMI-8**(6): p. 679-698.
46. Unser, M., A. Aldroubi, and M. Eden, *B-spline signal processing. I. Theory*. IEEE Transactions on Signal Processing, 1993. **41**(2): p. 821-833.
47. Schafer, R.W., *What Is a Savitzky-Golay Filter? [Lecture Notes]*. IEEE Signal Processing Magazine, 2011. **28**(4): p. 111-117.
48. Taha, A.A. and A. Hanbury, *An Efficient Algorithm for Calculating the Exact Hausdorff Distance*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. **37**(11): p. 2153-2163.
49. Slakter, M.J., *A Comparison of the Pearson Chi-Square and Kolmogorov Goodness-of-Fit Tests with Respect to Validity*. Journal of the American Statistical Association, 1965. **60**(311): p. 854-858.
50. Martin Bland, J. and D. Altman, *STATISTICAL METHODS FOR ASSESSING AGREEMENT BETWEEN TWO METHODS OF CLINICAL MEASUREMENT*. The Lancet, 1986. **327**(8476): p. 307-310.
51. Cheimariotis, G.A., et al., *ARCOCT: Automatic detection of lumen border in intravascular OCT images*. Comput Methods Programs Biomed, 2017. **151**: p. 21-32.
52. Gurmeric, S., et al. *A New 3-D Automated Computational Method to Evaluate In-Stent Neointimal Hyperplasia in In-Vivo Intravascular Optical Coherence Tomography Pullbacks*. in *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2009*. Springer Berlin Heidelberg.
53. Moraes, M.C., D.A.C. Cardenas, and S.S. Furui, *Automatic lumen segmentation in IVOCT images using binary morphological reconstruction*. BioMedical Engineering OnLine, 2013. **12**(1): p. 78.
54. Celi, S. and S. Berti, *In-vivo segmentation and quantification of coronary lesions by optical coherence tomography images for a lesion type definition and stenosis grading*. Medical Image Analysis, 2014. **18**(7): p. 1157-1168.