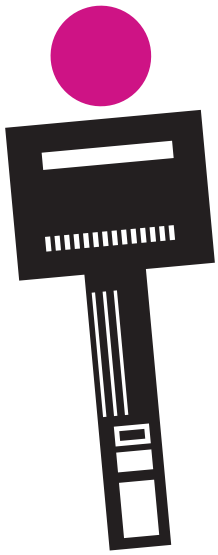


RAM ROBOTICS AND MECHATRONICS

VISION-BASED GUIDANCE FOR ROBOT ASSISTED ENDOVASCULAR INTERVENTION

A. (Alfred) Schell

MSC ASSIGNMENT

Committee:
prof. dr. ir. S. Stramigioli
dr. G. Dagnino
dr. ir. W.M. Brink

December, 2021

075RaM2021
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

Cardiovascular diseases are the predominant factor of death in the European Union. Over the years, a lot of emphasis have been put on minimal invasive procedures as they outperform the classical open surgeries. Robotic platforms have been designed to assist surgeons, in order to increase accuracy, repeatability, comfort, and post-operatory recovery times, but there are still disadvantages to be addresses. One of the main disadvantages is the imagining system based on X-ray angiography, which exposes both the patient and the practitioner to radiation. Another drawback is the lack of haptic feedback, such intervention requiring a skilled surgeon.

In this thesis, a new robotic platform for endovascular interventions - CathBot (developed at the Imperial College London) - is presented and a vision-based guidance system is developed to offer visual and haptic feedback to the operator, providing information on the interactions between vasculature and the whole catheter during surgery. The imaging system proposed works on X-ray angiograms and with the help of state of the art machine learning algorithms it is able to detect both the blood vessels and the surgical instruments. The proposed network architecture was able to detect the vasculature with an average accuracy of 94%, and the instrumentation with an average accuracy of 88%. The system achieved 4 frames per second on a mid-end machine. A framework for contact point detection and force estimation is also proposed.

Contents

1. Introduction 1
1.1 Imaging technologies 1
1.2 Endovascular robotics 1
1.3 Problem description and goals 2
1.4 Outline 3
2. Background 4
2.1 Related Work 4
2.2 CathBot 5
2.3 CNNs 5
2.4 ResNet architecture 6
2.5 U-net 8
3. Methodology 9
3.1 Dataset pre-processing 9
A. Data collection 9
B. Pre-processing 9
C. Data augmentation 11
3.2 Network architectures 12
A. U-net architecture 12
B. Siamese U-Net architecture 13
3.3 Loss functions 14
A. Binary Cross-Entropy 14
B. Dice loss 15
C. Combo loss 15
3.4 Contact points detection 15
4. Results 18
4.1 Phantom 18
4.2 Catheter 19
5. Discussions and future work 22
5.1 Research question 1 22
5.2 Research question 2 22
5.3 Research question 3 23
5.4 Future work 23
Navigation system 23
Mesh generation and FEA 23
6. Conclusion 25
Appendix A. Technical software 26
Appendix B. More results 27
Bibliography 30

1. Introduction

According to a study conducted in 2017 [1], cardiovascular diseases remain the main cause of mortality in nearly all EU member states, accounting for 37% of all deaths across EU countries. Vascular surgery is the clinical discipline concerned with the diagnosis and treatment of disorders of the arterial, venous, and lymphatic systems, exclusive of the intracranial and coronary arteries [2]. Endovascular surgery, as a minimally invasive procedure to treat a vascular disease from inside of a vessel via a remote site, is a special "area of interest" within the wide field of vascular surgery. This procedure is an alternative to open surgery and offers many advantages including smaller incisions, less trauma and a faster recovery for patients, and the possibility of using local anesthesia rather than general anesthesia [3].

Endovascular surgery is performed by manipulating guidewires and catheters through the vasculature in order to reach areas of interest where treatment is needed (e.g., stenting, ablation, embolization, device delivery). These procedures require only a small incision, through which the thin catheter is inserted. Using image guidance (X-Ray angiography, Magnetic Resonance Angiogram), the catheter is guided through a blood vessel to perform the required treatment [4].

1.1 Imaging technologies

X-ray imaging is usually used to detect solid structures like bones, but angiograms can be also acquired by injecting a contrast dye into the bloodstream, before taking the X-ray. The contrast agent is visible on the X-ray images and, as it moves through the blood vessels, the general shape, structure, and flow of the vessels can be identified. The ability of fluoroscopy to show catheters and guidewires also allows physicians to precisely manipulate the devices in real-time [5]. Despite the fact that both the practitioner and the patient are exposed to X-rays, this method is used because of its ability to provide real-time information. The advancements made in x-ray technologies lead to flat-panel detectors to be used, which reduce radiation exposure to patients undergoing interventional procedures, while enhancing clarity.

Magnetic Resonance Imaging (MRI) has also a great potential in diagnosing vascular diseases, offering detailed soft-tissue information without ionizing radiation. One disadvantage of MR scanners is the inability to perform in real-time, but the advancements in the last two decades (i.e., better hardware with rapidly switching and stronger magnetic gradients [6]), made MR-guided interventions possible. Other disadvantages are the need of MR-safe devices and instruments (i.e., that are electrically non-conductive, non-metallic, and non-magnetic) and that some patients can not undergo an MR procedure (i.e., they have certain implants or pacemakers).

1.2 Endovascular robotics

Despite the presented advantages, endovascular procedures are difficult to perform due to reduced sensory feedback, misalignment of visuo-motor axes, and the need for a high technical skill [7]. To overcome this limitations, teleoperated robotic navigation systems are used in the clinical workflow to increase dexterity and precision, reduce vessel wall contact, reduce operator's exposure to radiations, and to offer an ergonomic working position to the clinician. A setback of these teleoperated robotic navigation systems is the lack of haptic or force feedback during wire, catheter, or device manipulation.

The Sensei X and Magellan (Auris Health, Mountain View, CA, USA) are among the robotic catheterization systems that have been employed in clinical practice. The Sensei robotic system has an outer sheath controlled by tendon drives, manipulated by a physician and is designed for accurate positioning, manipulation and stable control of catheter and catheter-based

technologies during cardiovascular procedures. The tendon drives can be controlled with a joystick or navigation buttons. The Magellan Robotic System, which has been re-engineered from The Sensei robotic system, uses a smaller outer guide catheter to navigate inside the vasculature under 2D fluoroscopy. One of the drawbacks of this system is the lack of the implementation of haptic feedback [8].

CorPath GRX (Siemens Healthineers, Erlangen, Germany), is a robotic-assisted platform for endovascular interventions, which includes more controlled and precise device manipulation especially after reaching the target site. Movements as guidewire and catheter advancement, retraction and rotation can be performed by joysticks. However, a physician is still needed for vascular access and for initial catheter guidance, and additional personnel are required for any device deployments. Two major limitations of this device are the device selection, which is limited due to the compatibility only with 0.014 guidewires, and the absence of a mechanism for providing force sensing or haptic feedback to the operator [9].

Having acknowledged the novelty and potential that these robotic systems can provide in terms of dexterity and precision during interventional procedures, major limitations remain, such as deployment time, lack of haptic feedback, and the use of 2D fluoroscopy imaging systems.

1.3 Problem description and goals

Research initiated at Imperial College London - the CathBot project [10] - aims at creating a remotely manipulated MR-safe robotic system for performing endovascular interventions. The intervention is performed with the help of a master device with integrated haptic feedback and the slave robot performs guidewires and catheters manipulation in multi-modal imaging environments (i.e., Fluoroscopy and MR). The system addresses some of the aforementioned limitations of commercial robotic platforms for endovascular procedures. To achieve such versatility the imaging software needs to adapt to the different environments and extract useful information about the interaction between instruments and vasculature. Based on the obtained information, haptic feedback will be provided both in the manipulator and on screen.

This project focuses on developing a novel navigation system for the CathBot robotic platform. The novel navigation system aims to track how the catheter is being manipulated inside the vasculature, in order to provide feedback to the surgeon. This is an important step towards safer manipulation, as the catheter could damage the blood vessels during manipulation [11]. Because the contact between instrument and environment could happen either at the catheter's tip or along its body, which could result in punctures or bruises, it is important to track the whole catheter.

Standard image processing algorithms used for shape or edge detection would give poor (i.e., slow processing, noisy, unreliable) results for such a complex task, compared to algorithms based on Neural Networks (NN) [12]. Therefore, for the detection task, machine learning will be employed as it offers more flexibility and lower computational times. The U-Net architecture [13] is proposed for segmenting the objects of interest. An extended research will be made in order to find the optimal depth of the encoder in order to achieve real-time capabilities. The proposed architectures for the encoder are based on ResNet [14] with different number of layers.

The results will be processed such that boundary information of the objects of interest is extracted and used to provide feedback about possible contact points between the catheter/guidewire and the vessels.

The goal of this research is to answer three main research questions:

- Which Convolutional Neural Network (CNN) architecture is suitable for performing segmentation of the vasculature and catheters/guidewires in fluoroscopic and MR images?
- What improvements could be made to ensure the CNN's capability of performing in real time?
- How can contact points be determined from a 2D image, in a 3D environment?

A further goal was to also estimate interaction forces between the catheter and the environment. However, such a complex task required more time than was available and a software capable of performing inverse FEA (Finite Element Analysis). Therefore, only the concept of how such a feature could be implemented is presented here.

1.4 Outline

In the introductory part, it was presented why it is important to improve endovascular procedures and how robotics was integrated in order to achieve better results, while increasing the comfort of both the patient and surgeon. The rest of this thesis is structured as follows. A theoretical chapter is proposed for a better understanding of the current state of the art, and of the concepts used. In the third chapter, the methodology used to reach the proposed goals is presented. This is followed up with a quantitative analysis of the obtained results. In chapter 5 the results are discussed and future work is proposed for improving catheter navigation. The final chapter summarizes the results and provides a closing note.

2. Background

2.1 Related Work

The technologies developed for a better manipulation of endovascular instruments are focused on remote navigation since they reduce the physician's exposure to X-rays, improve catheter stability and reproducibility of the procedure, and increase the patient's safety by minimizing the contact and friction forces between catheter and blood vessels. Computer vision is also employed when using remote navigation techniques in order to enhance visibility and make the steering of the catheter easier, safer and faster. The advances made in the field of computer vision and machine learning combined with hardware improvements, made convolutional neural networks to be used for segmentation of biomedical images [15] [16] [17], with increase speed and accuracy.

Convolutional neural networks (CNNs) have become the state of the art when it comes to computer vision tasks [18], as they are designed to learn spatial hierarchies of features, automatically. In the beginning, CNNs were employed mostly in classification tasks, where the output of an image was a label specific to one class, but nowadays they are also used for image segmentation [19] and speech recognition [20].

In the medical field, image segmentation is used, given the need of spatial localization (i.e., each pixel is labeled, instead of the whole image). In general, the segmentation architecture is composed of two components, an encoder and a decoder. The role of the encoder is to extract feature maps and is usually a pre-trained classification network. The decoder's task is to project the discriminative features, learnt by the encoder, onto the pixel space. The main drawback of this technique is that by propagating through several convolutional and pooling layers (in the encoder), the feature maps are down sampled and the decoder retrieves the spatial information at lower resolution. To overcome this problem, Ronneberger et al. developed U-Net [13], a CNN architecture especially for biomedical image segmentation. The particularity of this architecture is that the decoder has a large number of feature channels in order for the network to propagate context information to higher resolution layers [13]. As a consequence, the encoder and decoder are symmetric and the whole architecture is U-shaped (Fig. 1). The proposed architecture was applied on three different biomedical segmentation challenges by ISBI (International Symposium on Biomedical Imaging). On the EM (Electron microscopy) segmentation challenge (2012) the U-Net architecture achieved the best score and on the cell tracking challenge (2014 and 2015) it achieved high IOU (intersection over union) scores of 92% and 77,5% on the “PhC-U373” and "DIC-HeLa" datasets, respectively.

In [21], U-Net was employed for polyp segmentation and was proven that the architecture has the ability to perform in real time, achieving 252 FPS (frames per second) for input images of 256 x 256 pixels. This is an important accomplishment, since it proves that the architecture could be used during surgeries. A big limitation of the U-Net architecture is that the optimal depth of the encoder is a priori unknown, leading to an extensive architecture research.

Another method for catheter and guidewire detection is proposed in [22]. The framework uses a multiscale vessel enhancement filter to increase the visibility of wire-like structures in X-ray images. Adaptive binarization is applied in order to extract the centerlines of wire-like structures and an algorithm is proposed for reconstructing the path of the targeted structures. KNN (k-nearest neighbors) classification is proposed for distinguishing between catheters and guidewires, and other wire-like artifacts. The detection rate achieved for catheters is between 91,4% and 84,8%, depending on the catheter's type, and the detection rate for guidewires is 83,5%. A frame rate of 11 FPS is achieved using this method.

2.2 CathBot

CathBot [10] is a novel teleoperated robotic platform for fluoroscopy and MRI-guided endovascular interventions. The system aims at addressing the following clinical requirements:

- Compatibility with different imaging modalities;
- Versatility: for performing a wide range of vascular interventions;
- State of the art navigation system: for minimizing contacts between the manipulated instrument and the vasculature;
- Teleoperation: for increased comfort of the surgeon;
- Usability.

The robot is designed as a master-slave device. The master device [23] is designed to mimic and map the established manual intra-procedural handling of standard catheters and guidewires. In this regard, the manipulator has a cylindrical handle that imitates the interaction and executable DOF (degree of freedom) (i.e., feeding/retraction and rotation) of standard catheters and guidewires. This design improves the teleoperation transparency, while keeping the controls intuitive for the user. A linear and a rotary brushless DC motors are used in order to provide haptic feedback for linear motion (feeding/retraction), and rotational motion respectively.

The slave robot is pneumatically actuated, has 4-DOF, and it can be used to perform different vascular interventions. It consists of two pneumatic linear stepper motors to translate the instrument, one pneumatic rotational stepper motor to rotate the instrument, and two pneumatic J-clamps to clamp while translating the instrument [23]. The slave's motion is based on the motion commands generated by the surgeon manipulating the master device. The parts of the slave robot are 3D printed from materials that are electrically non-conductive, non-metallic, and non-magnetic, making the robot MR-Safe.

2.3 CNNs

A well-known deep learning architecture applied to analyze visual imagery is the Convolutional Neural Network (CNN), a class of Artificial Neural Networks (ANN). The CNN architecture is composed of three types of layers: convolutional layer, pooling layer, and fully connected layer.

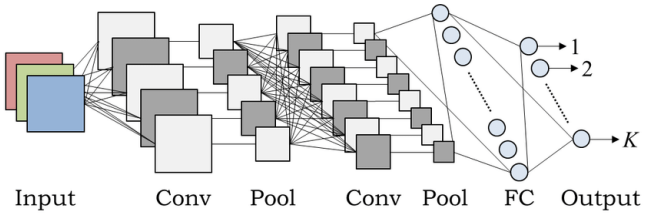


Fig. 1 : An example of a CNN architecture *[36]*

The convolutional layer is the core building block of the CNN. In this layer, a filter is applied to the input (the original image or a feature map) and a feature map is outputted. It works by performing dot product between two matrices, the first matrix being the set of learnable parameters (the kernel) and the second one being a portion of the receptive field (the size of the region in the input that produces the feature [24]). During the forward pass, the kernel slides across the height and width of the image or feature map resulting in a 2D matrix of features for each input channel.

The convolutional layer is the component with the most parameters in a CNN: number of kernels, size of kernels, activation function, stride, and padding. The *number of kernels* represents how many feature maps are outputted, while *size* represents the width and height of the applied kernel (usually 3x3, 5x5, 7x7, with bigger sizes working for some particular cases). The depth of the kernel is equal to the number of the input channels. *Stride* represents how many pixels the kernel slides at each step, while *padding* refers to the amount of pixels added to an image's border. The *activation function* has the role to determine if a given node in the network sends a signal to subsequent layers. There is a multitude of different activation functions: Sigmoid, Leaky ReLU, tanh, ReLU, Maxout, etc.; but the Rectified Linear Unit (ReLU) is the most widely used as it shows better convergence [25] . The advantage of this function is that it does not activate all neurons at the same time, making it more computationally efficient.

Pooling layers reduce the feature map size without loss of information. This helps in reducing the spatial size of the representation, which decreases the processing required further in the CNN, saving time and resources. The pooling operation consists in passing a filter over the feature maps and combining the pixel values inside the filter into a single value. The output value is dependent on the type of pooling applied: Max pooling (returns the max value inside the cluster), Average pooling (returns an average value of the pixels), or Sum pooling (returns the sum of all elements). This layer is characterized by the parameters: kernel size (usually 2x2), and stride.

Fully connected layers are used in classification tasks, and they represent a feedforward neural network. In this layer, the neurons have a complete connection to all the activations from the previous layers.

2.4 ResNet architecture

Residual Network (ResNet) is a specific ANN model that was introduced by *He et al.*, in [14]. This model became one of the most popular and successful deep learning models so far. Typical ResNet models are implemented with double or triple layer skips that contain nonlinearities (ReLU) and batch normalization in between [14]. By stacking additional layers in the deep neural network, we can solve complex problems with improved accuracy and performance, as these layers progressively learn more complex features.

Skip connections (as presented in ***Fig. 2***) are the essence of residual blocks. Having a direct connection which skips some layers simplifies the network and reduces the impact of the vanishing gradient problem (i.e., there are fewer layers to propagate through), leading to increased learning speeds.

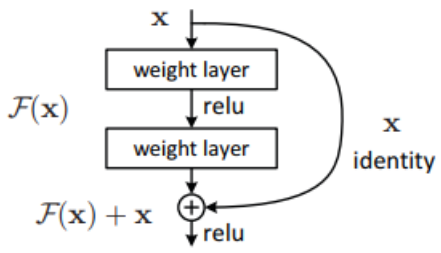


Fig. 2 : Residual learning: representation of a basic building block with skip connection *[14]*

The ResNet architecture is based on VGG (Visual Geometry Group) nets [26]. By adding the skip connections at each pair of 3x3 filters, the base network turns into its counterpart residual version. The detailed architectures are presented in ***Table 1***.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1 : Architectures for ImageNet. Building blocks are shown in brackets with the numbers of blocks stacked. Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2. Table from *[14]*

2.5 U-net

U-Net is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg [13]. It was developed based on the Fully Convolutional Neural network (FCN), and consists of a contracting path (the encoder) and an expansive path (the decoder). The network architecture is illustrated in ***Fig. 3***.

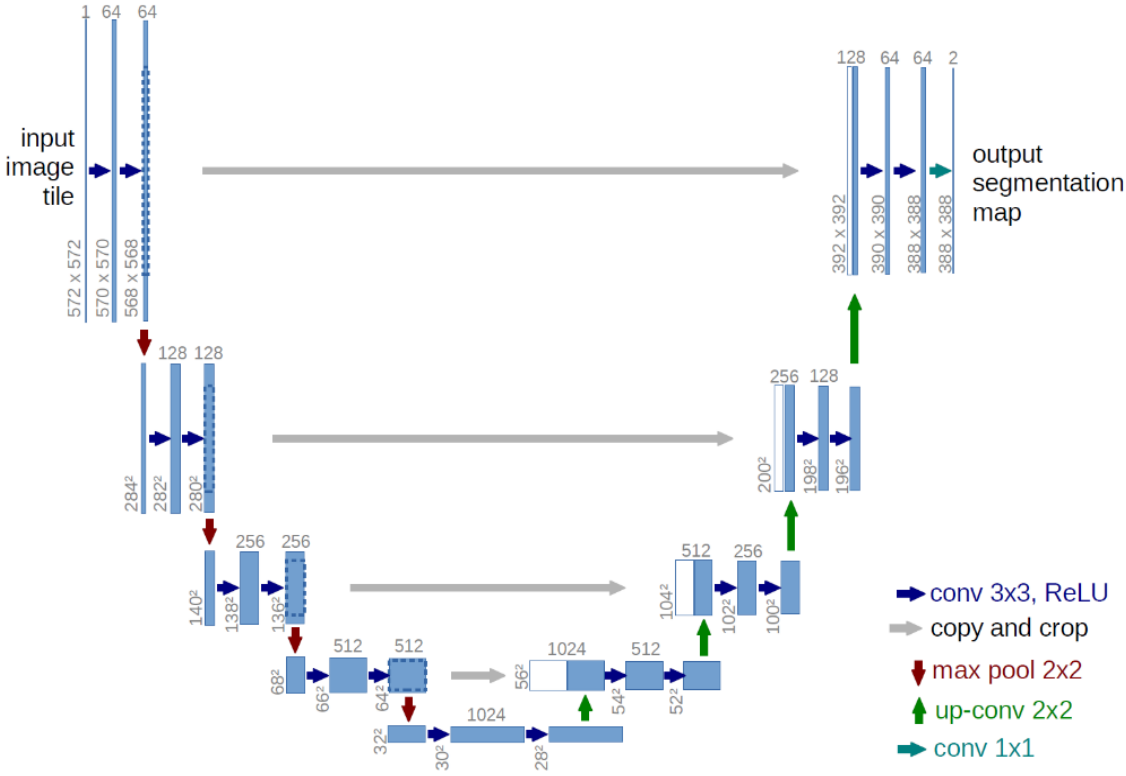


Fig. 3 : U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. *[13]*

The encoder is a typical architecture of a convolutional network. The particularity of this network is the decoder, as it combines the feature maps through a sequence of up-convolutions and concatenations with the correspondingly high-resolution feature maps from the encoder. This method ensures that the output will have a high resolution.

The U-Net architecture provides several advantages for segmentation tasks:

- It can be trained on a small dataset, which is very important in the biomedical domain, as datasets can be hard to acquire;
- It is trained end-to-end, thus the full context of the input image is preserved, because the image is processed entirely in the forward pass and segmentation maps are produced directly;
- Suited for segmentation tasks, as it computes a pixel-wise output;
- Is computationally efficient.

3. Methodology

The final scope of this project is to detect catheters and blood vessels in X-ray fluoroscopy images, in order provide visual and haptic guidance to the operator. The project is split into two parts.

In the first part, deep learning methods will be employed for detecting the vasculature's contour and the catheter, as it was proved they could achieve results fast and with high accuracies. As this project is meant to be integrated into a robotic platform for endovascular interventions, speed is an important factor when it comes to blood vessels and catheter detection, as the robot should receive such information in real time. Towards this end, a comparison between different neural networks, based on the U-Net architecture for both segmentation tasks, will be made.

The second part will focus on the feedback provided to the tele-operator. An algorithm will be developed for detecting the contact points between the catheter and the blood vessels.

3.1 Dataset pre-processing

A. Data collection

The dataset was composed of 45 videos of endovascular procedures conducted on phantoms, recorded at 30 Hz. The simulations were designed to represent a patient lying on the angiography table, with the help of vascular phantoms (Elastrat, Geneva, Switzerland) made from a soft silicone under an X-ray imagining system. To account for all possible procedures, multiple phantoms were used, representing the iliac; superior mesenteric; right common carotid and renal arteries. To improve the level of realism, the phantom was connected to a pulsatile pump to simulate normal human blood flow, and the cannulation procedures were conducted by four experienced vascular surgeons for each model.

Real-time video streams of the surgical scene were acquired using an image grabber (DVI2USB3, Epiphan Video, Ottawa, Canada) from the fluoroscopic system for interventional radiology procedure (Innova 4100 IQ GE Healthcare). The video stream was acquired on a workstation (Widows 10, Intel i7-6700HQ, 2.6GHz, 16GB RAM) and digitalized into image sequence for image processing (***Fig. 4***) [27].

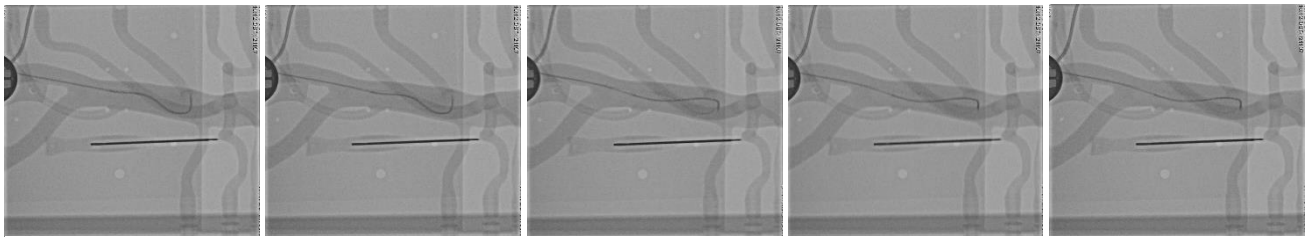


Fig. 4 : An example of 5 frames extracted from a recorded video, showing the manipulation of a catheter in the abdominal aorta.

B. Pre-processing

The obtained dataset consists of over 21.000 images of 1920 x 1200 pixels. Considering that CNNs requires the inputs (images) to have the same size, and large images occupy more space in the memory and require larger networks, our dataset needs to be resized to a smaller resolution. Choosing a smaller resolution comes as a tradeoff between computational efficiency and accuracy, because in the shrinking process the image features and patterns could be deformed. The training will be deployed on the GPU, as CPUs perform worse when it comes to deep learning methods [28]. The workstation's GPU memory is 4Gb and the models used have 22.397.289 (ResNet18) and 32.505.449 (ResNet34) parameters, meaning that the images can be resized to 512 x 512 pixels when batches of 2 are used. Furthermore, the images will be

converted to grayscale from RGB, as the nature of images does not offer relevant color information.

Downsizing could be applied by means of cropping or scaling down using interpolation. Both methods have the disadvantage of losing information, namely cropping can result in features or patterns removed near the border areas, while scaling can deform features or patterns if the scale ratio is not preserved. Since deforming patterns is less likely to affect accuracy, than losing them, our dataset will be downsized by means of bilinear interpolation. In ***Fig. 5***, we can see the deformed shape resulted in the resizing process. However, this deformation should not be a problem for the model's accuracy, as all images will be deformed in a similar manner, regardless of the phantom used.

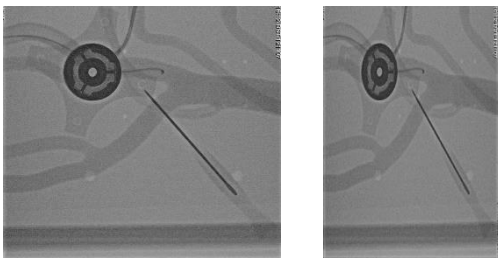
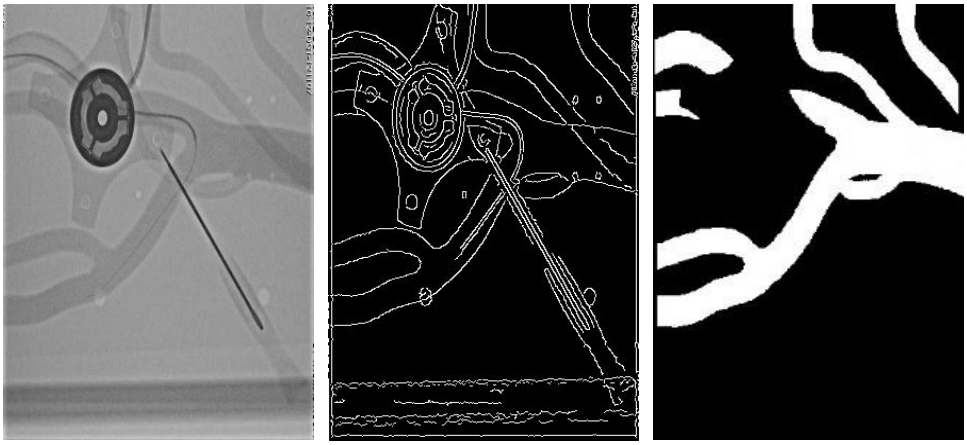


Fig. 5 : Deformation resulted from the resizing process
(left: The original image; right: The resized image)

In order to segment blood vessels and catheters in X-ray fluoroscopy images, ground truth labels need to be generated for training and testing purposes. The ground truth labels will be generated by assigning a proper label to each pixel (i.e., 0 for background, 1 for foreground), in a semi-automatic manner. Firstly, an algorithm will be used in order to highlight the edges of the interest object, and then little details will be refined manually with the help of a photo editing software (Gimp).

The blood vessels' contours were generated using the Canny edge detection algorithm, after noise has been reduced by applying Gaussian blur to the image. ***Fig. 6*** presents the steps of generating the ground truth labels for the blood vessels. Given the high similarity between the images of the same phantom model, only 100 ground truth images were generated for each class, exception being the RCC phantom for which only 35 ground truth images were generated. The dataset was split in proportion of 80% as training data and 20% as validation data. For testing purposes, another set of 10 images from each class were segmented.

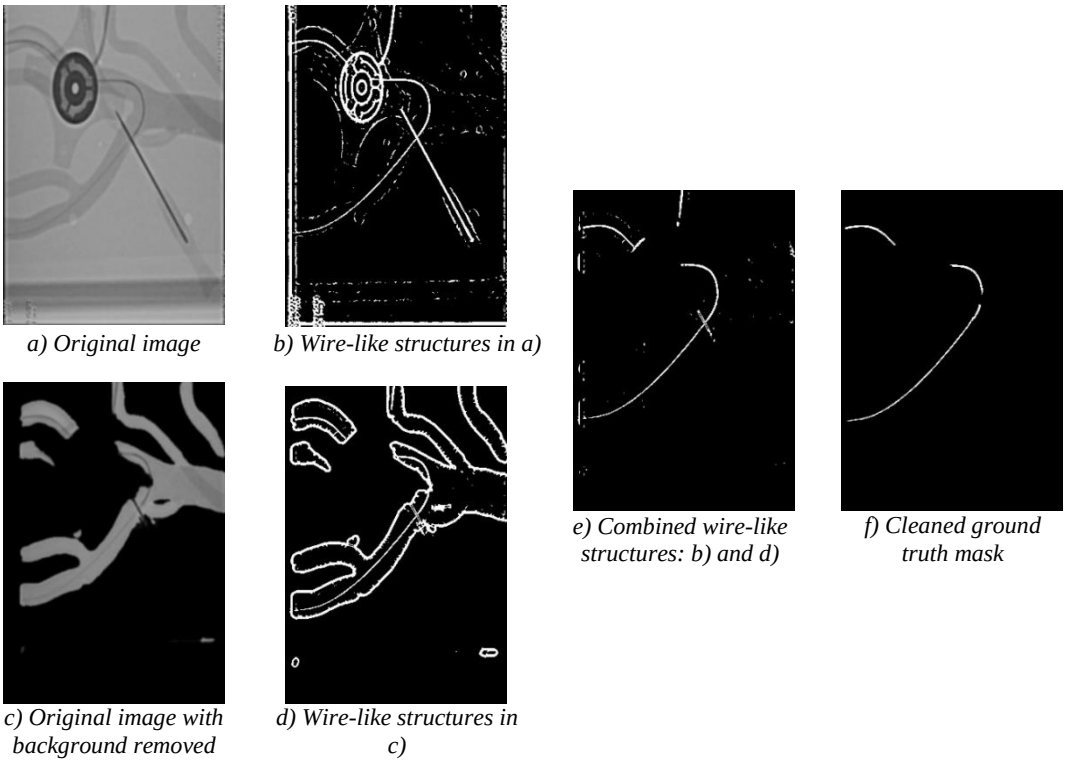


a) Original image *b) Detected edges* *c) Ground truth mask*

Fig. 6 : Ground truth generation pipeline for the blood vessels: From the original image a) the edges are detected b) and the final mask c) is manually edited

Generating the catheter's ground truth labels was achieved by using the base method proposed in [22] with some modifications. The image was smoothed with the help of a Gaussian filter, then a 2 x 2 Hessain matrix for each pixel was formed and decomposed. Two eigenvectors and eigenvalues were computed for each matrix (i.e., pixel) and the eigenvalues were arranged in ascending order. The eigenvalues that were smaller than 0.01 were filtered out, the remaining eigenvalues representing wire-like structures. In this step, it was observed that there are a lot more wire-like structures in our images, besides the catheter, which generated a lot of noise in the segmented image ***Fig. 7,* b)**. To overcome this problem, the same algorithm was applied again on the image, but this time the background has been removed (i.e., the ground truth labels generated for the blood vessels were used as masks on the original images) ***Fig. 7,* d)**. The two binary images created were then combined with the logical operator "AND" ***Fig. 7,* e)**. The remaining noise was removed manually with the help of a photo editing software (Gimp).

The resulted dataset consists of 688 images, divided as follows: 114 - Iliac; 255 - SMA; 160 - LR; 159 - RCC. The dataset was divided into 85% training data and 15% validation data (~103 images). For the testing set, 25 random images were selected (which were not used in the training phase) for each class.



a) Original image *b) Wire-like structures in a)* *e) Combined wire-like structures: b) and d)* *f) Cleaned ground truth mask* *c) Original image with background removed* *d) Wire-like structures in c)*

Fig. 7 : Ground truth generation pipeline for the catheter: From the original image a) and with the background removed c), wire-like structures are extracted b) and d). The results are then combined with the binary operator "AND" e). Finally, the mask f) is manually cleaned.

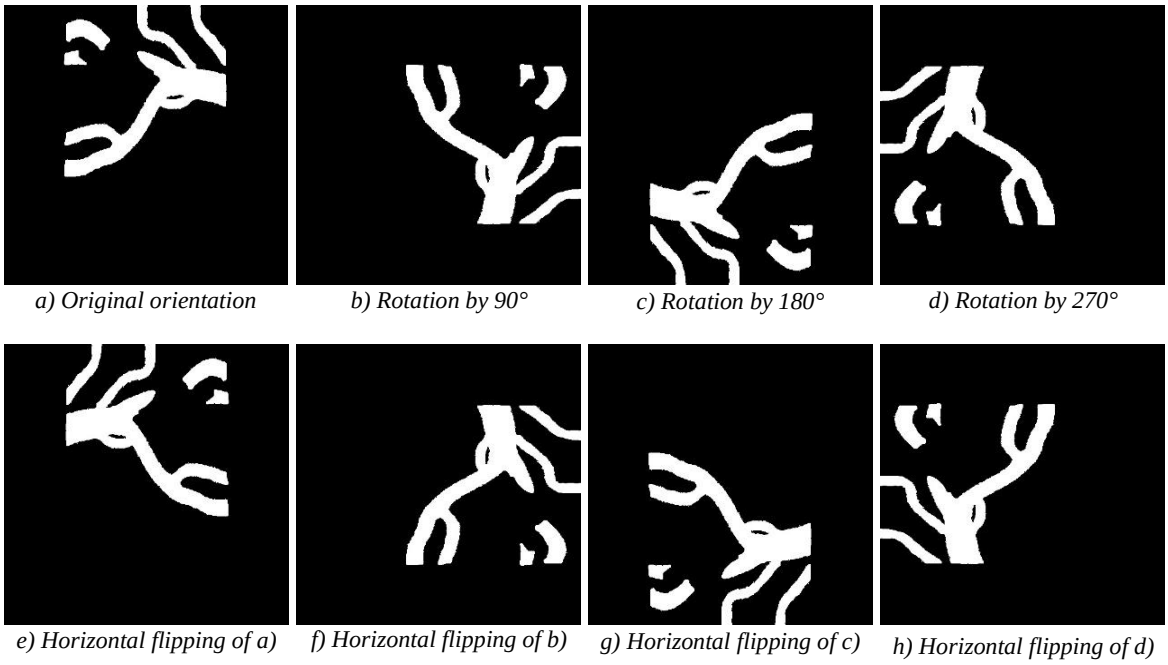
C. Data augmentation

Deep learning applications require a lot of data in the training phase. Several methods have been employed in order to reduce the needed data while preserving accuracy, such as using pre-trained CNN models, utilizing regularization methods (i.e., dropout or data augmentation) [29]. For a bigger flexibility of the trained models, in this project data augmentation will be used. This process will be done online (i.e., transformations applied directly on batches, during the training process). For this purpose, the following transformations: Flipping - horizontal; Rotation - by 90, 180 or 270 degrees.

Flipping the images ensures that more positions are considered, especially in the catheter's segmentation case, since the catheter can move freely during maneuvering. In the case of blood vessel segmentation, horizontal flipping will improve the model's perception about the general shape of the vessels, with the added benefit of making the model ready for the rare cases of situs inversus.

Rotations were applied in steps of 90 degrees. This ensures that the model will segment the object of interest, regardless of the imaging system orientation.

Because the augmentation is done on-line, the transforms are applied randomly and independently, with a chance of 50% for each. There is a chance of 25% for both transforms to be applied. If rotation is applied, there is an even chance for each degree to be applied. The transforms are applied to both input image and the mask in the same manner, in order to retain the labels' values of truth. All possible outcomes are presented in ***Fig. 8***.



a) Original orientation *b) Rotation by 90°* *c) Rotation by 180°* *d) Rotation by 270°*

e) Horizontal flipping of a) *f) Horizontal flipping of b)* *g) Horizontal flipping of c)* *h) Horizontal flipping of d)*

Fig. 8 : Possible outcomes of the augmentation method

3.2 Network architectures

Real time segmentation of blood vessels and catheters in X-ray sequences is required for our application, which insinuates that the deep learning model should have a high accuracy while performing fast. The segmentation model will rely on the U-Net approach, since it was proven to have high accuracies for biomedical data [13], and the convolutional part will be based on a classical CNN architecture focused on reducing the computational costs.

To extensively investigate the encoder's architecture on performance, two pre-trained networks will be implemented and compared for both models. Even more, in the case of catheter segmentation, the temporal information will be taken into account by employing the Siamese U-Net approach.

A. U-net architecture

The U-Net architecture consists of two streams: a convolutional (encoder) and de-convolutional (decoder) path. The convolutional path will be based on a pre-trained (on ImageNet dataset [30]) ResNet architecture [14] with 18 and 34 layers, presented in ***Table 1***. The final FC layer of the ResNet architecture was removed as only the 5 building blocks are needed to extract depth features from the input images. After each ResNet layer, a map is

outputted with the size of 256, 128, 64, 32, and 16 respectively. These maps are obtained in the first layer, conv1, from a 2x2 max pooling operation and by performing down-sampling in the subsequent layers conv3_1, conv4_1 and conv5_1, respectively.

In the decoder path, each block consists of an up-sampling of the feature map by means on bilinear interpolation, a concatenation with the corresponding feature map copy from the encoder path, and a 3x3 convolution followed by a ReLU. In the last layer of the expansive path, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes [13].

The complete architecture is presented in ***Fig. 9***.

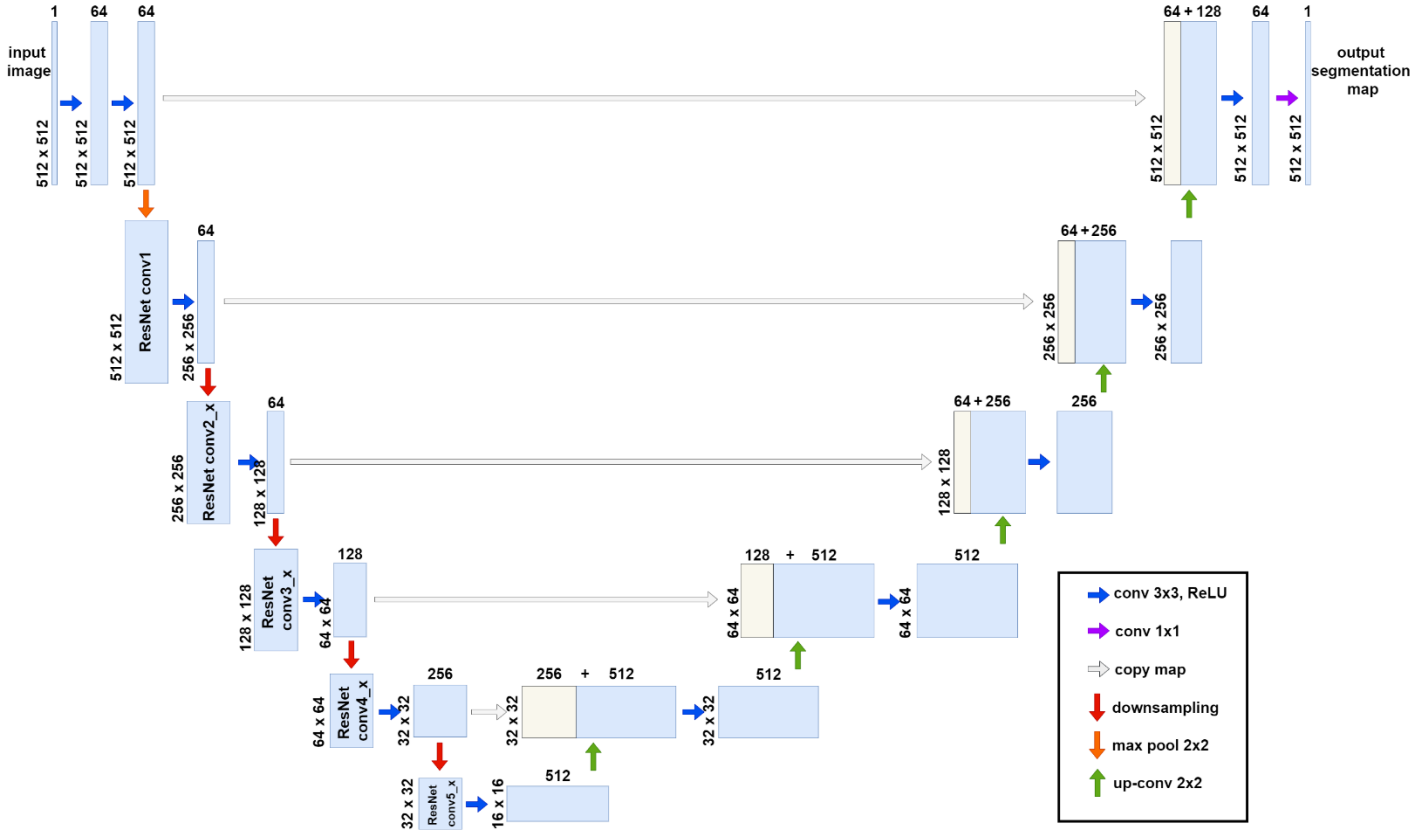


Fig. 9 : The proposed U-Net architecture

B. Siamese U-Net architecture

The proposed architecture for catheter segmentation that also considers the temporal information is based on a Siamese variant of the U-Net architecture. The encoder will be based on the ResNet architecture, as in the previously discussed U-Net method. The main difference is that in this configuration, instead of one encoder, two identical encoders with shared weights will be used. Similarly, the decoder will be modified to accustom the concatenation of three feature maps. The complete architecture is presented in ***Fig. 10***.

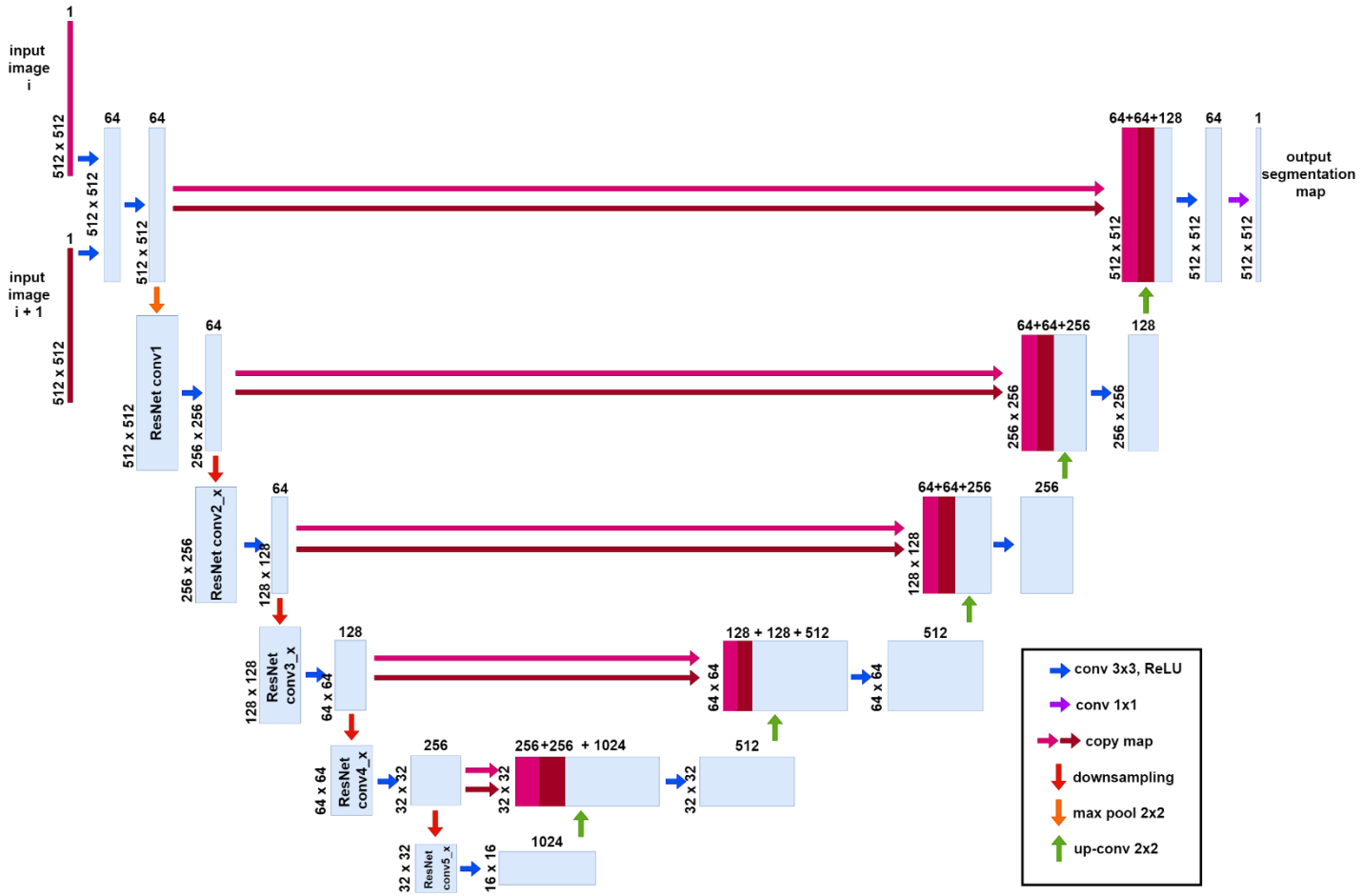


Fig. 10 : The proposed Siamese U-Net architecture

3.3 Loss functions

A. Binary Cross-Entropy

Cross-entropy [31] is a measure for calculating the difference between two probability distributions for a given random variable or set of events. It is widely used in classification tasks, in both binary and multi-class problems. Considering that the segmentation map of all our networks is labelling pixels as wither background (0) or foreground (1), the Binary cross-entropy loss function will be considered for training optimization.

Binary Cross-Entropy is defined as:

$$L_{BCE}(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

Where $\hat{y}$ is the model's output (i.e., the predicted scalar value), and y is the target value (i.e., ground truth label).

The BCE loss accepts only inputs between 0 and 1, otherwise the logarithms of $\hat{y}$ and (1-$\hat{y}$) would not exist. In this case, the only compatible activation function with the BCE loss is the Sigmoid.

B. Dice loss

Dice loss [32] is a popular loss function for image segmentation tasks when ground truth is available. It is based on the Dice coefficient, a widely used metric in computer vision community to calculate the similarity between two images, which is essentially a measure of overlap between two samples. Dice loss can be expressed as:

$$L_D = \frac{2pg + \varepsilon}{p + g + \varepsilon}$$

Where p is the predicted pixel's value, g is the ground-truth pixel's value and ε is a smoothness factor that ensures stability by avoiding the numerical issue of dividing by 0 (i.e., p and g =0). In this project, ε =1.

In order to have a loss function which can be minimized, the soft Dice loss will be used, which is 1-L_D.

C. Combo loss

The combo loss [33] is the weighted sum of Dice loss and BCE loss.

By analyzing both datasets, it is clear that the blood vessel class account for 70-80% of the pixels, while the catheter for only ~10%. Due to the class imbalance, the Dice loss is preferred (i.e., we need to segment a small foreground from a large background). But, to make training smoother, BCE is preferred. In order to combine both of their advantages, we will use a weighted sum of them, defined as:

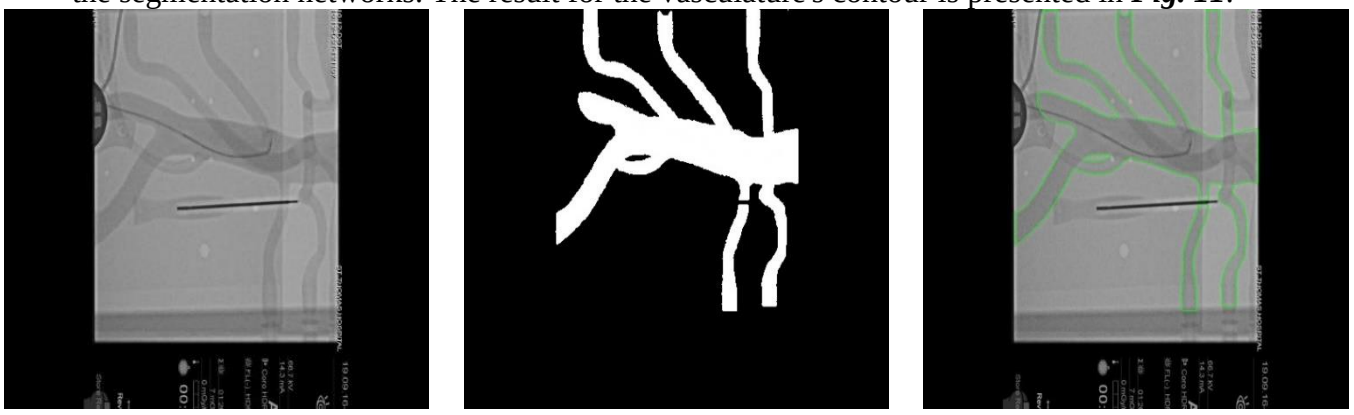
$$L = \alpha L_D + (1 - \alpha) L_{BCE}$$

Where α represents the weight of the Dice loss, which was varied between 0.6 and 0.1, in order to find the best ratio.

3.4 Contact points detection

Contact points between catheter and vasculature can be defined as the intersection between the two's contours or when they are in close proximity to one another. We can assume a cylinder to be a simplified model of a blood vessel (**Fig. 15**). Considering that our imaging system provides 2D images, there is no depth information available about the catheter's position relative to the symmetry plane of the cylinder. Therefore, the contact points will be determined based on proximity, rather than contours intersection, in order to avoid injuries.

For contour retrieval, the algorithm [34] will be applied on the binary images retrieved by the segmentation networks. The result for the vasculature's contour is presented in ***Fig. 11***.



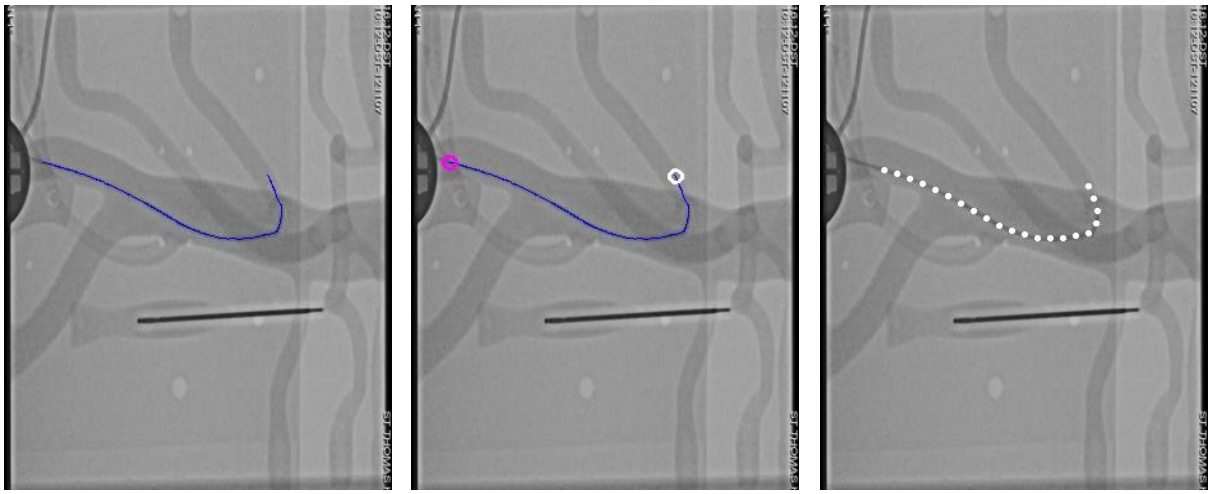
a) Original image *b) Vasculature map* *c) Extracted contour (green) interpolated in the original image*

Fig. 11 : Contour retrieval workflow: From the original image a) the vessel map is segmented b) and based on that information the contour is extracted c)

Finding the catheter's contour is similar to the previous case. Given the fact that the catheter is rather thin (4-6 pixels wide) it is safe to simplify its contour to a single line, passing right through the middle (***Fig. 12***). This method saves computational resources without loosing accuracy. On the obtained shape a filter is applied that finds the end points, in order to determine where the catheter's tip is positioned (***Fig. 13***, ***b***). Further simplifications are made by sampling the line in equal segments (***Fig. 13***, ***c***). The sampling frequency should be chosen such that the points would define small segments that could accurately represent the initial shape.

a) Original segmentation *b) Thinned segmentation*

Fig. 12 : Thinning process for the catheter (zoomed in)



a) Detected catheter contour *b) Detected end points (white: Tip; pink: Base)* *c) Sampled points from the catheter's contour*

Fig. 13 : End point detection and sampling: From the extracted contour a) the end points b) and sample points c) are extracted.

Proximity will be determined using a method similar to ray casting [35]. For all points determined along the catheter (i.e., the sampled points), a normal ray will be cast. When the ray intersects the blood vessel's contour, distance between them will be computed. If the distance is smaller than a given contact threshold (in pixels), the ray's origin will be considered to be in contact with the blood vessel (***Fig. 14***).

The contact threshold is introduced in order to compensate for the lack of depth information. As the environment is perceived as two dimensional, we do not know with certainty where the catheter is positioned inside the vessels, with respect to the third dimension (depth). For this method to work, the pixel spacing and the focal distance attributes are required. Depth will be estimated by comparing the catheter's measured diameter to its real diameter (provided in the technical data sheet). A straight line starting from the vessel's center pointing downwards with the length of the height (*h* in ***Fig. 15*** **b)**) will be considered to be a side of a triangle. Another side, the hypotenuses, will start from the center with length equal to the vessel's radius (*r* in ***Fig. 15*** **b)**) as to form a right triangle. By applying the Pythagorean Theorem we can find the length of the last side of the formed triangle. The contact threshold will be defined as the difference between the vessel's radius (*r*) and the found value (*l* in ***Fig. 15*** **b))**.

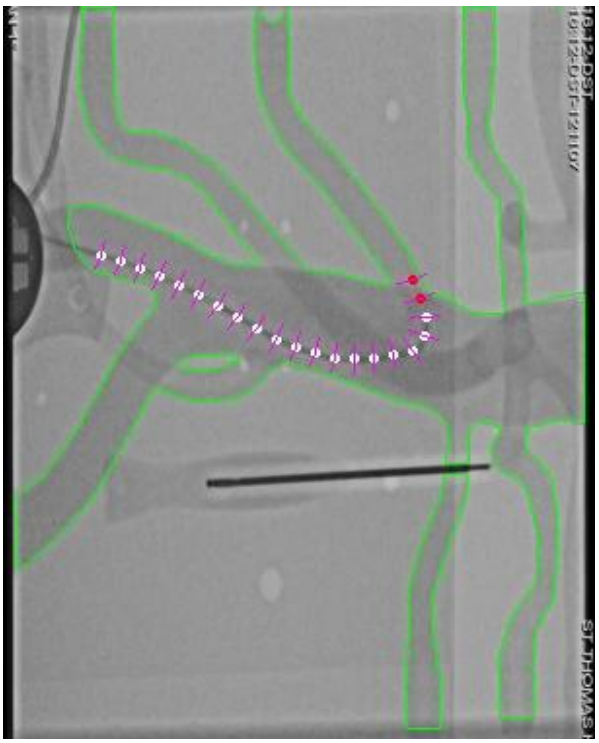
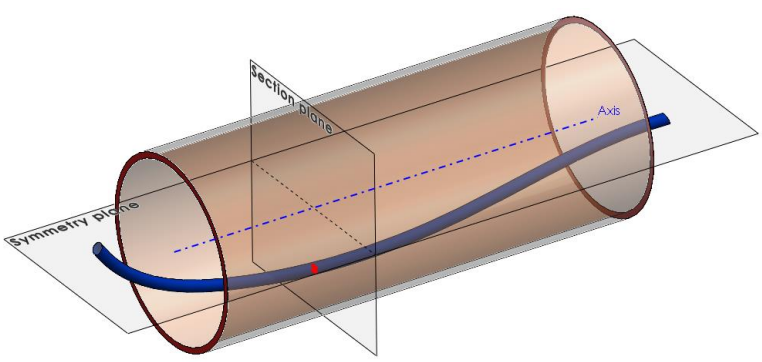
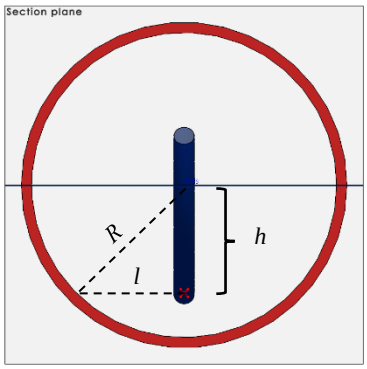


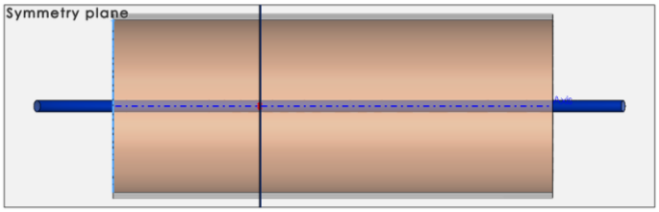
Fig. 14 : Ray casting for contact detection: A normal ray (represented in pink) is casted from each sampled point (represented in white). If the ray intersects the vasculature's contour, a contact point is determined (represented in red).



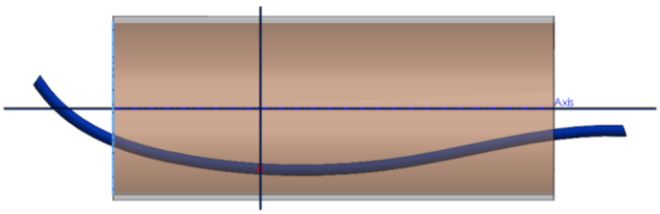
a) Perspective view



b) Section view



c) Top view



d) Side view

Fig. 15 : Simplified model of the blood vessels. The passing catheter (blue) is situated under the symmetry plane of the vessel, which also represents the focus plane of the fluoroscope. The top view c) represents the information provided by the imaging system. The section view b) demonstrates why proximity is preferred over contact detection and how the contact threshold (CT) is computed. The perspective and side views are presented for orientation purposes.

4. Results

In this section the training and testing results will be presented for the various architectures implemented.

We evaluate the segmentation accuracy of the X-ray images considering the degree of overlap between the predicted regions and the regions of the ground truth. In our experiment, the dice metric was used to quantitatively evaluate the segmentation.

4.1 Phantom

The architectures used for segmenting the blood vessels were based on U-Net.

For the following architectures, the encoder was based on ResNet18. For training, a batch size of 2 was used for 40 epochs, and the Adam optimizer was applied with a learning rate of 0.001 and a learning rate decay of 0.1 after 30 epochs. Model 1 has the Combo coefficient $\alpha = 0.5$, while for the second model $\alpha = 0.6$, prioritizing the BCE outcome. Models 2, 3 and 4 prioritize the Dice loss more and more, with a Combo coefficient $\alpha = 0.4$, $\alpha = 0.3$, and $\alpha = 0.2$ respectively.

Architecture	Combo coef. α	Best Epoch	Max validation loss		
			BCE	Dice	Combo
Model 1	0.5	24	**0.073277**	0.090236	**0.081756**
Model 2	0.6	22	0.131683	0.121089	0.127445
Model 3	0.4	35	0.106915	0.096473	0.100650
Model 4	0.3	37	0.112288	**0.086603**	0.094309
Model 5	0.2	28	0.118579	0.090907	0.096442

Table 2 : Combo loss coefficient influence on the validation loss for a U-Net architecture (Bold represents the best result).

Table 2 summarizes the best validation loss achieved for the blood vessels segmentation. In ***Fig. 16*** and ***Fig. 17*** the training and validation loss, respectively, are illustrated across all 40 epochs.

Fig. 16 : Validation loss evolution for the models presented in ***Table2***.

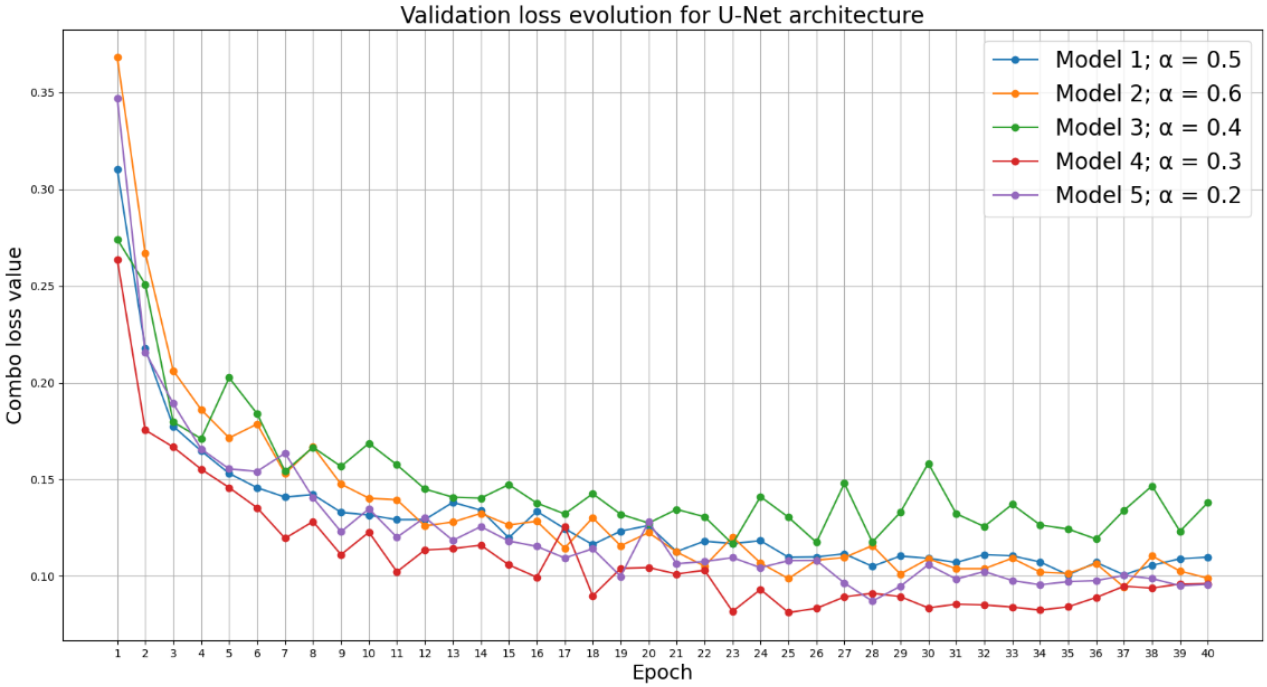


Fig. 17 : Validation loss evolution for the models presented in ***Table2***.

Testing was done for all the architectures using the dataset specified in B. Pre-processing. Only accuracy (measured with the Dice metric) was tested, as the framerate (frames/second) can't vary too much for the same architecture. ***Table 3*** presents the obtained results.

Model	Dice score (average)
Model 1	0.944141
Model 2	0.922497
Model 3	0.939202
Model 4	**0.944424**
Model 5	0.941709

Table 3 : Blood vessel segmentation - testing results (Bold represents the best result).

4.2 Catheter

The architectures used for segmenting the blood vessels were based on U-Net (for model 1 and 2) and Siamese U-Net (for model 3 and 4) with the Combo loss coefficient α=0.5. Table 4 summarizes the best validation loss achieved for the blood vessels segmentation.

Architecture	Epoch	Max validation loss		
		BCE	Dice	Combo
Model 1 (U-Net, ResNet18)	39	0.004093	0.085873	0.044983
Model 2 (U-Net, ResNet34)	39	0.003242	0.084096	0.043669
Model 3 (Siam U-Net, ResNet18)	33	0.004449	0.086243	0.045346
Model 4 (Siam U-Net, ResNet34)	40	0.003572	0.081732	0.042652

Table 4 : Best training validation loss.

The evolution in time (epochs) of the training and validation phases for the U-Net architecture can be seen, for each model, in ***Fig. 18***:

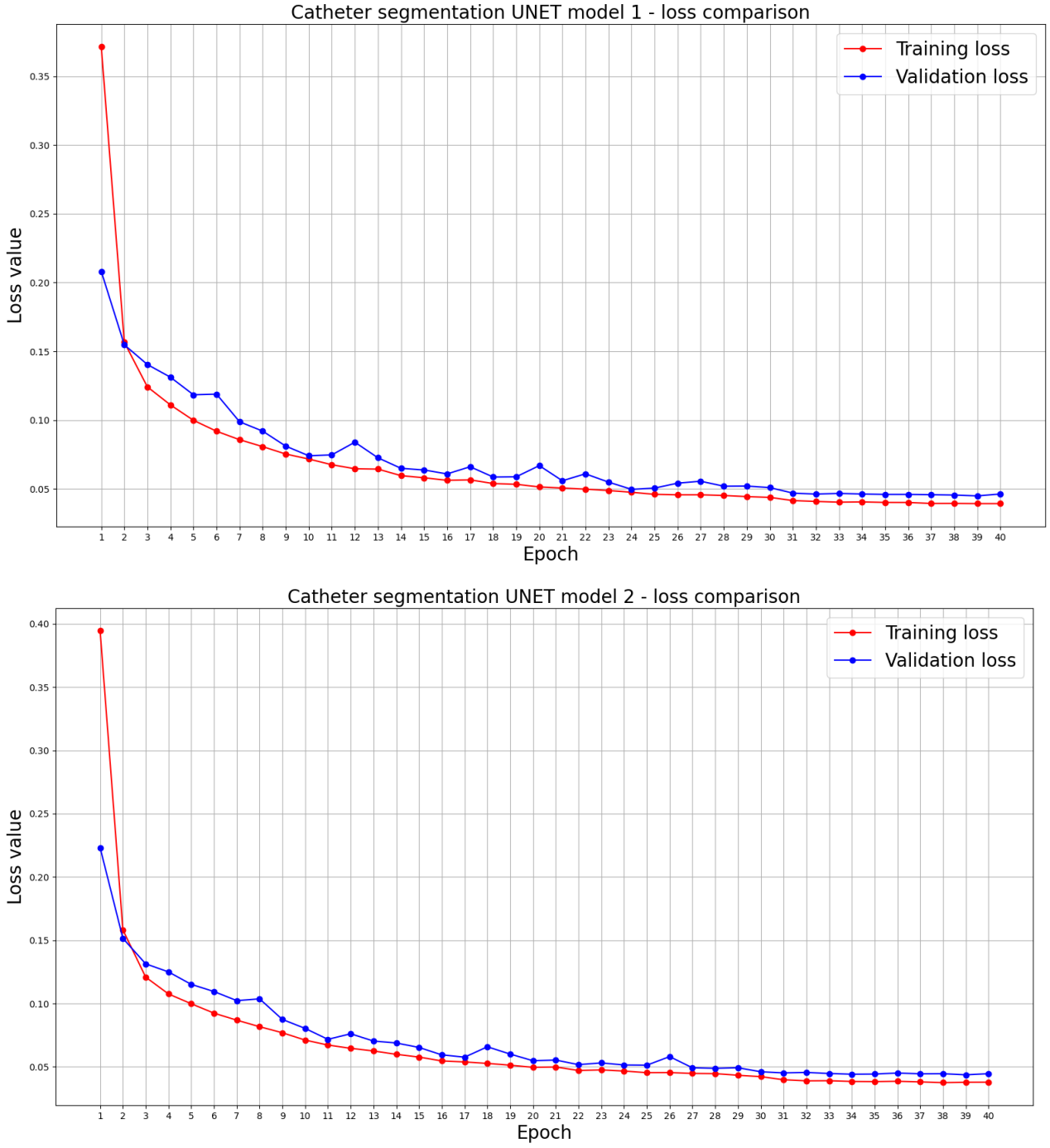


Fig. 18 : Loss evolution for the U-Net architecture.

The training and validation evolution of the models based on the Siamese U-Net architecture can be visualized in ***Fig. 19***:

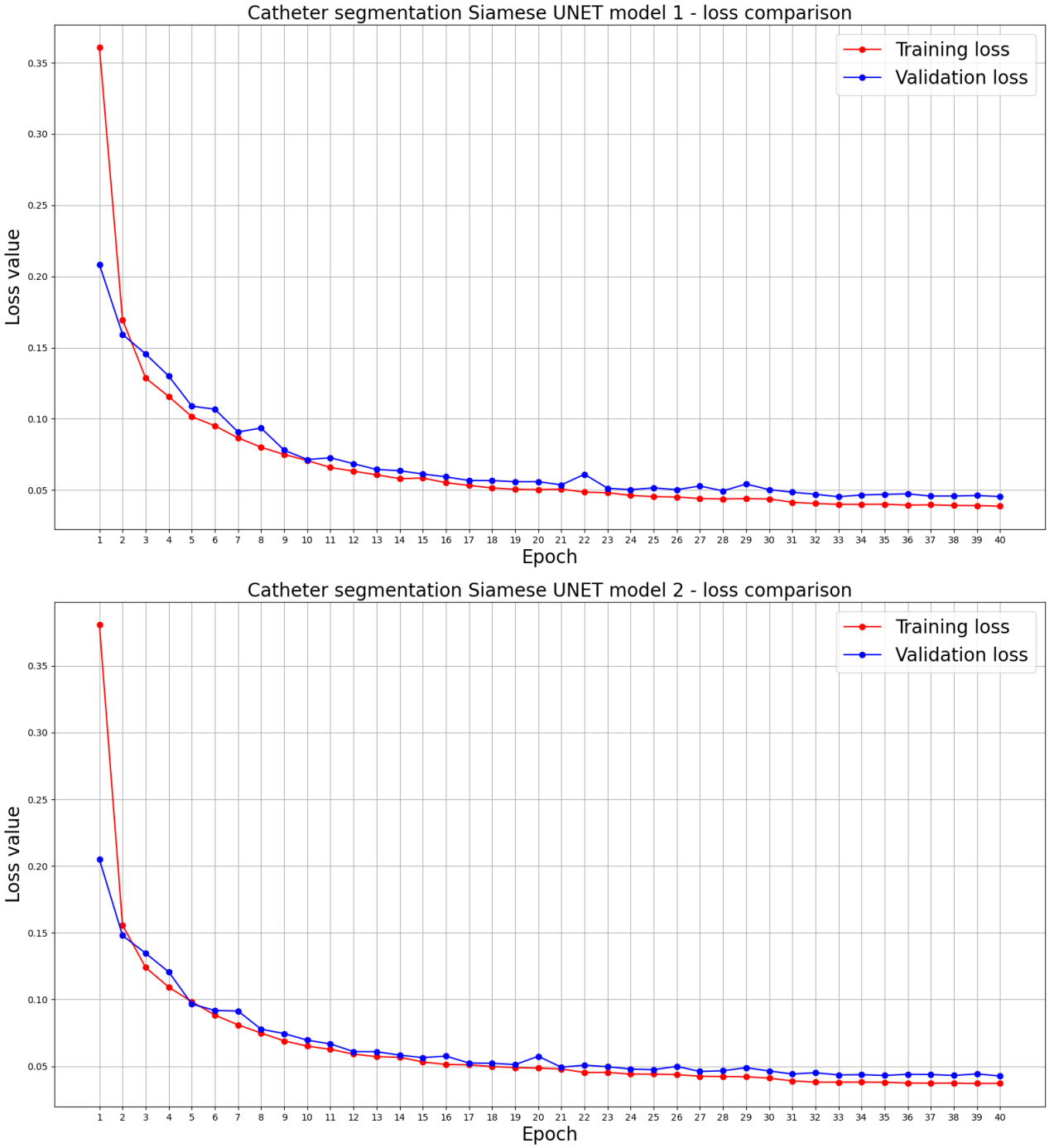


Fig. 19 : Loss evolution for the Siamese U-Net architecture.

The testing results are presented in the following table:

Model	Dice score (average)	FPS
Model 1 (U-Net, ResNet 18)	0.874435	1.986
Model 2 (U-Net, ResNet34)	**0.882244**	1.828
Model 3 (Siam U-Net, ResNet18)	0.807976	2.808
Model 4 (Siam U-Net, ResNet34)	0.811457	**2.933**

Table 5 : Catheter segmentation - testing results

5. Discussions and future work

In this section the results obtained will be discussed and the research questions will be answered.

5.1 Research question 1

Which CNN architecture is suitable for performing segmentation of the vasculature and catheters/guidewires in fluoroscopic and MR images?

Analyzing the results obtained, it is clear that the U-Net architecture is suited for performing segmentation for both catheters and vasculature. In the case of the blood vessels, U-Net has very high accuracies (over 90%) while obtaining an average speed of 4.8 FPS.

In ***Table 3*** the difference between architectures with a Combo loss with different factors α are compared, for vessel segmentation. From the difference between the results, we could conclude that Dice loss should be more important than the BCE loss when it comes to segmentation. For values of $\alpha \leq 0.5$ the difference in results is very small and this could be attributed to the randomness introduced by the data augmentation techniques. Anyway, the big difference in performance between $\alpha = 0.6$ and $\alpha \leq 0.5$ leads us to believe that any factor higher than 0.5 will lead to poor results. As model 1 has the best accuracy, for the moment a factor of $\alpha = 0.5$ is recommended and used for the catheter segmentation models.

For catheter segmentation, the U-Net architecture still performs well, with accuracies of up to 87%. The accuracy is lower than in the previous case due to the high-class imbalance (i.e., there are much more black pixels than white ones). By analyzing the model's outputs, it can be observed that the low accuracy is due to the inability of fully detecting guidewires, which are smaller than catheters. From ***Table 5*** we can conclude that by using a deeper encoder we can increase the accuracy, but not by much. Comparing the classical U-Net implementation with the Siamese U-Net architecture, it is clear that the classical implementation can provide higher accuracies at lower speed. In this case, a trade-off between accuracy and speed should be made when choosing the right model for segmentation.

5.2 Research question 2

What improvements could be made to ensure the CNN's capability of performing in real time?

From the obtain results it can be concluded that the implemented models do not achieve real-time capabilities. One decisive factor for the low number of frames processed each second is the used GPU. A newer model could decrease the computational times leading to an increase of FPS.

The imaging software is designed to detect blood vessels and catheters for each frame. To make the process faster parallelization should be used. Another way of speeding the vasculature detection could be to perform the segmentation only once in multiple frames. Vasculature segmentation should be performed constantly because the system needs to adapt to the changes that occur due to breathing and heartbeat. Therefore, the updated vasculature could be segmented based on these measurable parameters, thus less often than every frame.

5.3 Research question 3

How can contact points be determined from a 2D image, in a 3D environment?

The developed framework for catheter detection was designed to account for depth information inside the vasculature. While the best solution would be to use a 3D imagining hardware, the proposed method is also useful if calibration is done right. The practitioner can see in real time if there are possible contact points or if the catheter is in close proximity with any vessel. The fact that this method tracks the whole catheter and not just the tip, is also a big advantage. Because the catheter's model is simplified for this task, the obtained points are also used in generating the mesh for FEA.

5.4 Future work

Navigation system

The obtained segmentation results are good, but catheter segmentation could be highly improved. Based on the obtained results, it is greatly recommended to increase the dataset and improve the ground truth labels, which could increase accuracy significantly. Other network architectures could be studied, especially the architectures that combine the spatial information with the temporal one. Even if the Siamese U-Net is supposed to account for that, it is clear based on the results from ***Table 5*** that it is not as accurate as a simple U-Net implementation, but it can perform faster. FW-net [27] was proposed for taking into consideration the temporal information, but in order to work, a very good ground truth flow is required. Toward this end, synthetic data is recommended to be created. Similarly to the flying chairs dataset [37], images with blood vessels and catheter could be generated with the help of a photo editing software.

The models were trained on data obtained in vitro. In order to achieve good results in vivo, the models, as they are, should be re-trained on real-life data. A drop in accuracy may be noticeable, as the real-life angiograms are noisier and have higher contrast.

The depth estimation algorithm does not work with variable depths along the catheter. It works by assuming a certain depth is kept along the catheter. Taking advantage of the sampled points, an improvement would be to estimate the depth of each point and generating a variable contact threshold accordingly.

Mesh generation and FEA

Force estimation could be achieved by doing an inverse FEA (Finite Element Analysis) as proposed in [36]. Due to time constraints and the lack of a good open-source software to perform such a task in real-time, force estimation remains a future scope of the project. Performing FEA requires a mesh of the object of interest, and it is considered a useful feature to be integrated in the current project, as the mesh should be constructed from image information. A simple framework for mesh generation is proposed as a stepping stone towards an automated FEA.

The concept behind the finite element model is that boundary conditions could be extracted from the catheter's shape in order to compute the needed force to achieve such deformations. The boundary conditions will be applied to a cantilever model of the catheter. In Error! Reference source not found., this concept is exemplified. From the imaging system, the contact points are extracted (in our case, we could use all the sampled points or only the detected contact points) and the cantilever model is computed. The undeformed catheter cantilever model is supposed to lay on the tangent line of the catheter with the length measured from base point to the tip. The imaging software could estimate the tangent line as the line formed of the base point and the next point along the catheter at a maximum distance of 10 pixels.

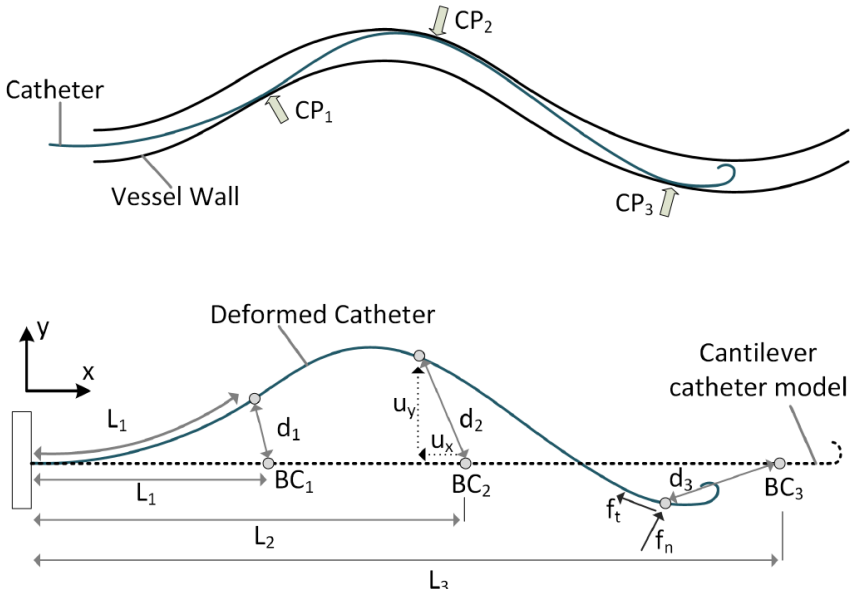


Fig. 20 : Force estimation concept using FEM model: CPi is the contact point, Li is the length of contact point on the catheter, BCi is boundary condition points on catheter model which is the corresponding point for CPi, di is the deflection of contact points, ux and uy are the components of deflection with respect to coordinate of cantilever. Image from [36].

The mesh will be generated by using information form the catheter's data sheet (i.e., diameter, material properties, etc.) combined with the extracted information about the contact points. With the sampled points extracted during contact detection, and knowing the base and tip points, an ordered array of points (from base to tip) could be generated. Because the points will be pretty close to each other, we could estimate the whole catheter length, and subsequently the cantilever's length, as the sum of distances between each two consecutive points. Inside the FEA software, the cantilever mesh should be generated based on the obtained information.

Extensive research has been made about FEA software that could be integrated in Python. Unfortunately, CAD (computer aided design) software such as *SolidWorks* and *Abaqus* have FEA capabilities but do not perform in real time. Other open-source software has been examined, such as *sfepy* and *FreeCAD* which have the advantage of being developed in Python and the potential of running the simulation in real time as the graphical interface could be turned off. Unfortunately, they do not have the capability of performing inverse FEA, but this feature could be integrated in the future.

6. Conclusion

The purpose of this thesis was to develop a vision-based guidance system for robot assisted endovascular interventions, as part of the CathBot (Imperial College London) platform. The guidance system here presented, is able to map the vasculature and detect the whole catheter, in order to assess the interaction between them and provide the operator with useful feedback about the procedure. An important achievement of this thesis is the ability to track the catheter along its whole length, as vessel damages can occur due to contacts along the entire catheter, and not only at the tip.

Machine learning algorithms have been employed for the segmentation task. The U-Net architecture was used with different encoder configurations. For the blood vessels, it was proven than even a simple ResNet-18 could achieve high accuracies, of up to 94%. For the catheter segmentation, the U-Net encoder was based on both ResNet-18 and 34, but a Siamese version was also tested, as it uses the temporal information as well. The accuracies achieved were over 80%. The guidance system has to work in real time, and this can be achieved by parallelizing the segmentation processes, and by using a powerful GPU.

The outputs of the segmentation steps have been processed to generate useful information about the interaction between the catheter and the vasculature. An algorithm for determining the contact points, which also takes into consideration the depth of the catheter has been developed. With the extracted information about the catheter's position, a mesh can be generated and used in a finite element analysis for estimating contact forces.

In conclusion, this thesis represents an important contribution towards better and safer robot assisted endovascular interventions.

Appendix A. Technical software

In this section, the software will be explained and how to use it.

Repository available at: shorturl.at/fszT6

The repository consists of different folders to provide an overview of the many files. The structure is as follows:

- *data*: Folder containing all the data that is or will be used by the main program.
- *functions*: This folder contains all the modules needed by the program.

1. Data folder structure

The data folder should contain all the needed files for running the main program. This folder should have the following structure:

- *Videos*: A folder containing all the video files; This folder will not be needed when a direct interface with DICOM will be developed;
- *Models*: A folder containing all the trained CNN models needed for segmentation;
- *Results*: If results need to be saved, this folder will be used.

2. Functions

As was specified, this folder holds all the modules needed by the program:

- *functions*: Contains all the functions needed to run the main program;
- *GUI*: Module for initializing and handling the user interface;
- *customModels*: Each model's architecture is defined here

A short description of each function available in the *functions* module is presented:

- *MapContour*: This function takes an image as input argument and returns the found contours inside it.
- *filterEndPoints*: This is the filter used for end-point detection. It takes a pixel as an input and returns 0 if the pixel is not an end-point or 255 if it is.
- *GetEndPoints*: Uses the filter *filterEndPoints* to find the end-points. It takes as an input the image over which we need to apply the filter. Returns a list of locations of end-point pixels.

3. User interface

The user interface (**Fig. 21**) consists of a small window that provides a better navigation and parameter setup. A search bar is provided for selecting the folder path where the videos are located. (Default path is to *Videos* folder). After the wanted folder has been chosen, a list of available videos is presented. The user can choose the video to be played by clicking on the desired name. When the video was selected, the program starts to run in the background. In the beginning, all the required CNN models are loaded into the memory and parameters are set. The Status bar beneath the video lists will provide the user with feedback about the current status and framerate. When initialization is done, a new window will be opened showing the video overlaid with the feedback about the environment.

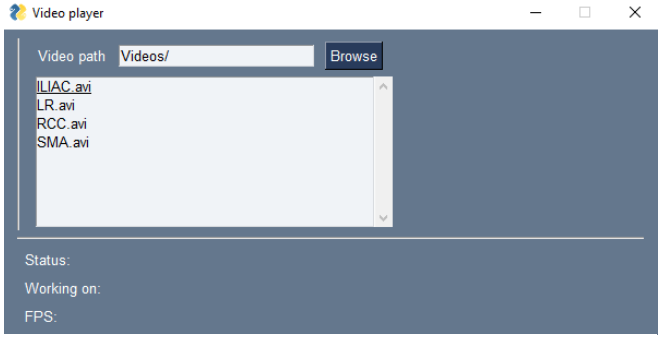
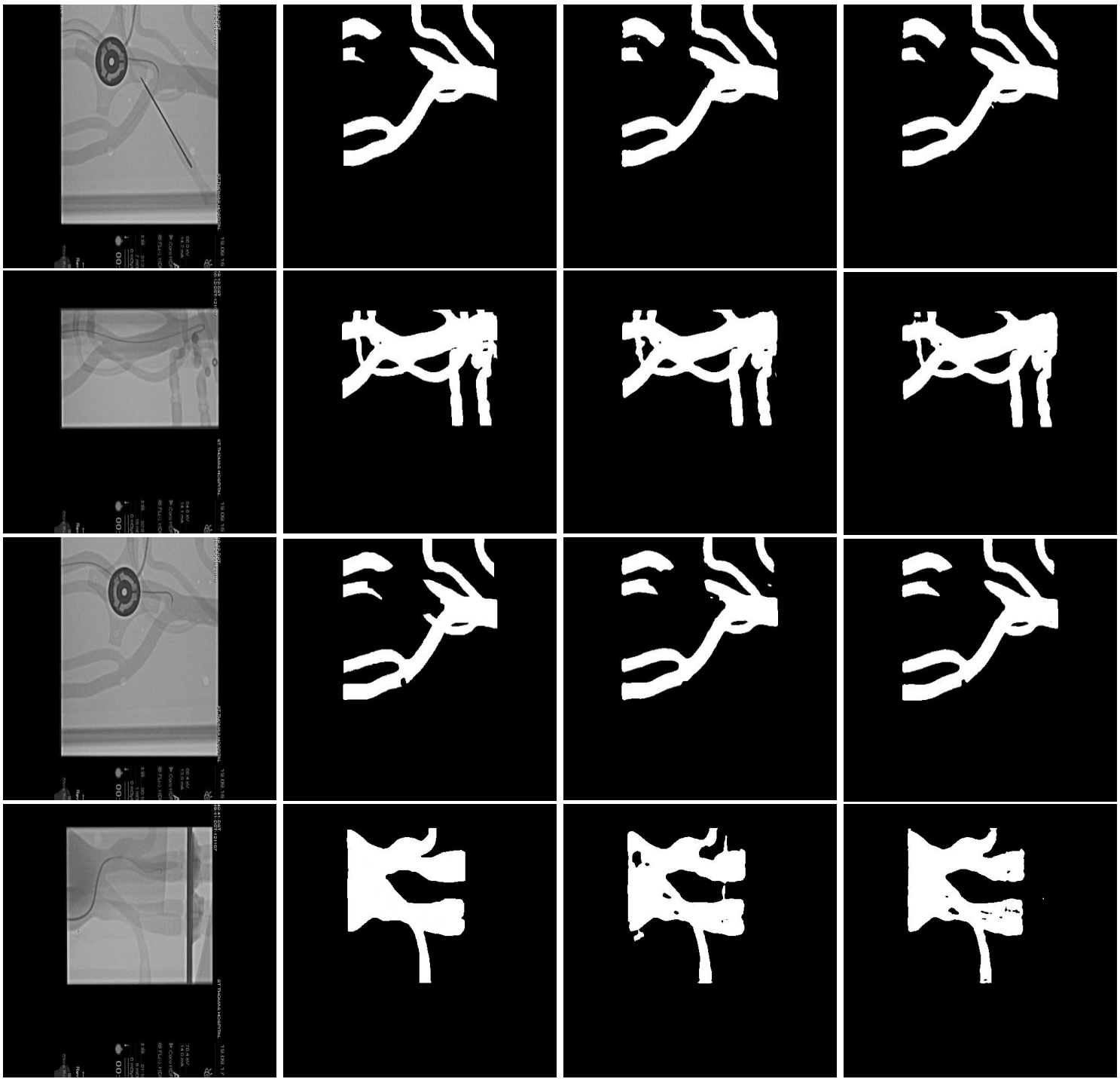


Fig. 21: The user interface

Appendix B. More results

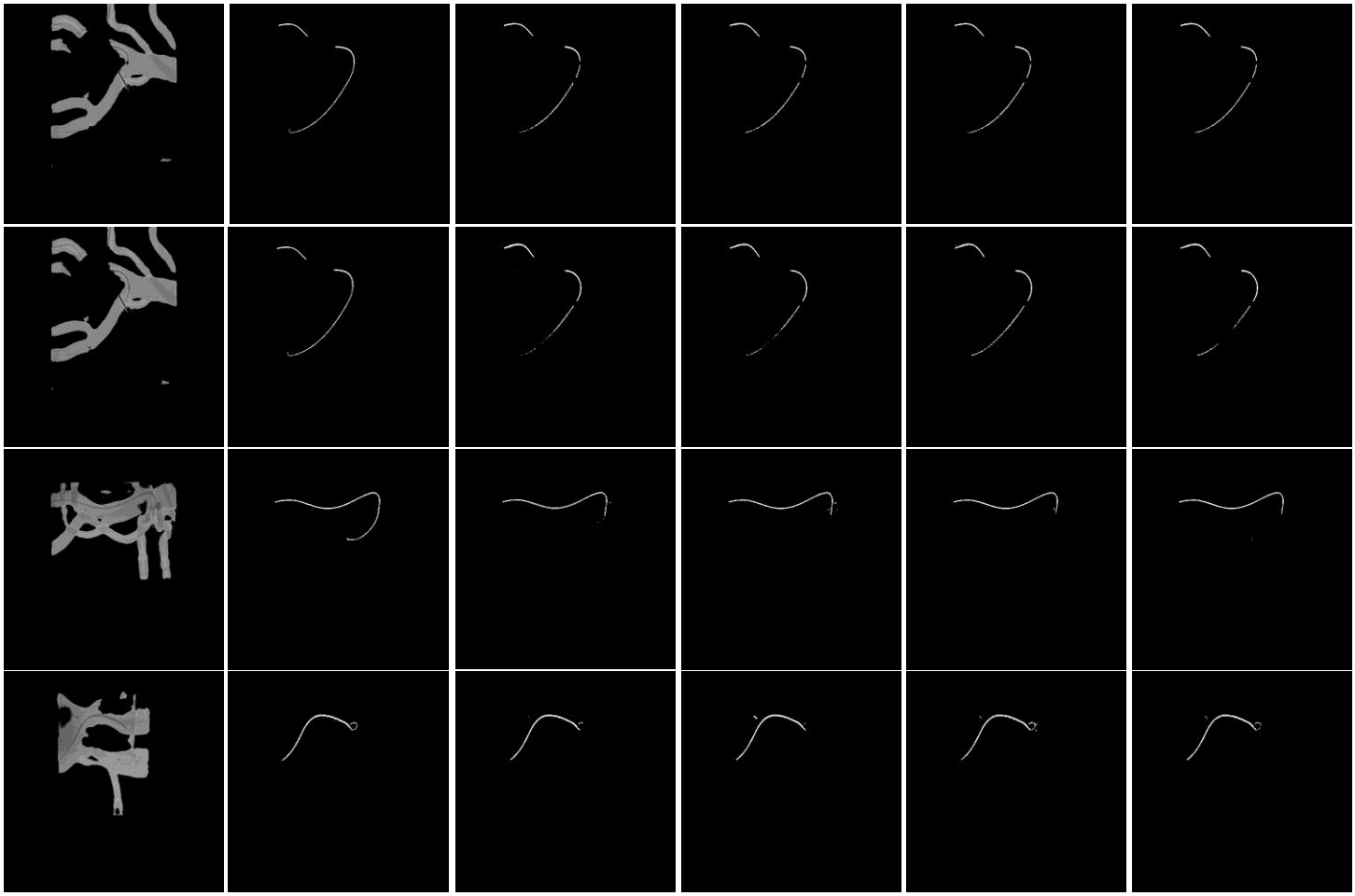
In this section some segmentation results will be shown.

For the blood vessel segmentation, 4 different cases are presented for the top 2 models:



Vasculature segmentation: First row - Original images; Second row - Ground truth mask;
Third row -Segmentation from model 1; Fourth row - Segmentation from model 4

Four different cases are illustrated for catheter segmentation:



Catheter segmentation: First row - Original images; Second row - Ground truth mask;
Third row -Segmentation from model 1; Fourth row - Segmentation from model 2; Fifth row - Segmentation from model 3;
Sixth row - Segmentation from model 4

Bibliography

[1] OECD and the European Union, "Health at a Glance: Europe 2020," 2020.

[2] Society for Vascular Surgery, "Definition of Vascular Surgery," n.d.. [Online]. Available: https://vascular.org/about-svs/definition-vascular-surgery. [Accessed 14 October 2021].

[3] UCSF Health, "Endovascular Surgery," n.d.. [Online]. Available: https://www.ucsfhealth.org/treatments/endovascular-surgery. [Accessed 14 October 2021].

[4] S. Christiansen, "What Is Endovascular Surgery?," 3 June 2021. [Online]. Available: https://www.verywellhealth.com/endovascular-surgery-5100836. [Accessed November 2021].

[5] W. Forrest, "Endovascular Imaging Options," 2015.

[6] C. Ozturk, M. Guttman, E. R. McVeigh and R. J. Lederman, "Magnetic Resonance Imaging–guided Vascular Interventions," *Top Magn Reson Imaging,* 2007.

[7] M. B. Molinero, G. Dagnino, J. Liu, W. Chi, M. E. M. K. Abdelaziz, T. Kwok, C. Riga and G. Yang, "Haptic guidance for robot-assisted edovascular procedures: Implementation and evaluation of surgical simulator," *IROS,* 2019.

[8] S. Gunduz, H. Albadawi and R. Oklu, "Robotic Devices for Minimally Invasive Endovascular Interventions: A New Dawn for Interventional Radiology," *Advanced Intelligent Systems,* vol. III, no. 2, 2020.

[9] P. Legeza, G. W. Britz, T. Loh and A. Lumsden, "Current utilization and future directions of robotic-assisted endovascular surgery," *Expert Review of Medical Devices,* vol. XVII, no. 9, 2020.

[10] M. E. M. K. Abdelaziz, D. Kundrat, M. Pupillo, G. Dagnino, T. M. Y. Kwok, W. Chi, V. Groenhuis, F. J. Siepel, C. Riga, S. Stramigioli and G.-Z. Yang, "Toward a Versatile Robotic Platform for Fluoroscopy and MRI-Guided Endovascular Interventions: A Pre-Clinical Study," *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 2019.

[11] J. Abram, J. Klocker, N. Innerhofer-Pompernigg, M. Mittermayr, M. C. Freund, N. Gravenstein and V. Wenzel, "Injuries to blood vessels near the heart caused by central venous catheters," *Anaesthesist,* no. 65, p. 866–871, 2016.

[12] G.-b. Xu, G.-y. Zhao, L. yin, Y.-x. Yin and Y.-l. Shen, "A CNN-based edge detection algorithm for remote sensing image," in *2008 Chinese Control and Decision Conference*, 2008.

[13] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Computer Science Department and BIOSS Centre for Biological Signalling Studies,* 2015.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[15] Y. P. Yanfei Guo, "BSCN: bidirectional symmetric cascade network for retinal vessel segmentation," *BMC Medical Imaging,* 2020.

[16] M. Gherardini, E. Mazomenos, A. Menciassi and D. Stoyanov, "Catheter segmentation in X-ray fluoroscopy using synthetic data and transfer learning with light U-nets," *Computer Methods and Programs in Biomedicine,* 2020.

[17] S. Yang, J. Kweon, J.-H. Roh, J.-H. Lee, H. Kang, L.-J. Park, D. J. Kim, H. Yang, J. Hur, D.-Y. Kang, P. H. Lee, J.-M. Ahn, S.-J. Kang, D.-W. Park, S.-W. Lee, Y.-H. Kim and Cheol, "Deep learning segmentation of major vessels in X-ray coronary angiography," *Scientific Reports NatureResearch.*

[18] Z. Li, W. Yang, S. Peng, F. Liu and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems,* 2020.

[19] S. B. Nemade and S. P. Sonavane, "Image Segmentation using Convolutional Neural Network for Image Annotation," *International Conference on Communication and Electronics Systems (ICCES),* pp. 838-843, 2019.

[20] V. Passricha, R. K. Aggarwal and R. Lopez-Ruiz, "Convolutional Neural Networks for Raw Speech Recognition," in *From Natural to Artificial Intelligence - Algorithms and Applications,* IntechOpen, 2018.

[21] G. Batchkala and S. Ali, "Real-Time Polyp Segmentation Using U-Net with IoU Loss," 2020.

[22] Y. Ma, M. Alhrishy, S. A. Narayan, P. Mountney and K. S. Rhode, "A novel real-time computational framework for detecting catheters and rigidguidewires in cardiac catheterization procedures," *Medical Physics,* 2018.

[23] G. Dagnino, J. Liu, M. E. M. K. Abdelaziz, W. Chi, C. Riga and G.-Z. Yang, "Haptic Feedback and Dynamic Active Constraints for Robot-Assisted Endovascular Catheterization," *International Conference on Intelligent Robots and Systems (IROS),* 2018.

[24] A. Araujo, W. Norris and J. Sim, "Computing Receptive Fields of Convolutional Neural Networks," 2019. [Online]. Available: https://distill.pub/2019/computing-receptive-fields/. [Accessed 20 November 2021].

[25] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks".

[26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR 2015*, 2015.

[27] A. Nguyen, D. Kundrat, G. Dagnino, W. Chi, M. E. M. K. Abdelaziz, Y. Guo, Y. Ma, T. M. Y. Kwok, C. Riga and G.-Z. Yang, "End-to-End Real-time Catheter Segmentation with Optical Flow-Guided Warping during Endovascular Intervention," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[28] F. B. Uz, M. Salvaris and D. Grecoe, "GPUs vs CPUs for deployment of deep learning models," 2018. [Online]. Available: https://azure.microsoft.com/en-us/blog/gpus-vs-cpus-for-deployment-of-deep-learning-models/. [Accessed 20 Octomber 2021].

[29] L. Brigato and L. Iocchi, "A Close Look at Deep Learning with Small Data," 2020.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision,* 2015.

[31] M. Yi-de, L. Qing and Q. Zhi-bai, "Automated Image Segmentation Using Improved PCNN Model Based on Cross-entropy," *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing* , 2004.

[32] F. Milletari, N. Navab and S. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *2016 Fourth International Conference on 3D Vision (3DV),* 2016.

[33] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu and G. Hamarneh, "Combo Loss: Handling Input and Output Imbalance in Multi-Organ," 2021.

[34] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing,* 1985.

[35] S. D. Roth, "Ray casting for modeling solids," *Computer Graphics and Image Processing,* 1980.

[36] M. Razban, J. Dargahi and B. Boulet, "A Sensor-less Catheter Contact Force Estimation Approach in Endovascular Intervention Procedures," *International Conference on Intelligent Robots and Systems (IROS),* 2018.

[37] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v. d. Smagt, D. Cremers and T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks," *IEEE International Conference on Computer Vision (ICCV),* 2015.

[38] A. Hidaka and T. Kurita, "Consecutive Dimensionality Reduction by CanonicalCorrelation Analysis for Visualization of ConvolutionalNeural Networks," in *ISCIE International Symposium on Stochastic Systems Theory and its Applications,* 2016.

[39] W. Zhu, Y. Huang, H. Tang and Z. Qian, "AnatomyNet: Deep 3D Squeeze-and-excitation U-Nets for fast and fully automated whole-volume anatomical segmentation," 2018.