

# Toolbox for vessel X-ray angiography images simulation

Gabriela Copetti Maccagnan<sup>1</sup>, Jean Schmith<sup>1,2</sup>, Marcia Santos<sup>1</sup>,  
Rodrigo Marques de Figueiredo<sup>1,2</sup>

<sup>1</sup>Polytechnic School, Unisinos University  
Av. Unisinos, 950, CEP 93022-00, São Leopoldo, Brasil

<sup>2</sup>Technology in Automation and Electronics Laboratory - TECAE Lab, Unisinos University  
Av. Unisinos, 950, CEP 93022-00, São Leopoldo, Brasil

{gabrielaacmacc, j.schmith, cunhasantosmarcia}@gmail.com, marquesf@unisinos.br

**Abstract.** *In recent years, automatic computer techniques have been proven to be a great tool for the rapid detection and disease diagnosis. The core of those diagnostic systems are usually artificial intelligent algorithms like convolutional neural networks, in which thousands of images are needed for training. However, the available datasets of biomedical images, specially for X-ray angiography, are scarce. Therefore, we propose a toolbox for X-ray angiography images simulation to increase the number of available images as an alternative to data augmentation method for training artificial intelligence algorithms. The toolbox was developed with a set of functions to simulate complex vessel structures, as well as stenosis and aneurysms, in X-ray angiography images.*

## 1. Introduction

Angiography is a procedure that, through radiography, captures the image of vessels that present contrast in the subject's circulation system. Among the applications of angiographic images are the examination of the kidneys, lungs and brain, however, heart analysis is possibly the most known usage of this procedure. In the heart case, the contrast is inserted into the vessel from an intravenous injection through the main coronary artery [Vijayalakshmi 2015]. Angiography images are used to analyse stenosis or abnormal aneurysms in the vessel wall, which may indicate the presence of a serious injury or a heart disease. This radiographic image is captured from an angle at a specific point of view, and it is often necessary to capture this event from different angles for a complete analysis of the vessel. As the contrast is injected and travels through the coronary vessel within a specific time frame, a video is recorded of that contrast flowing through the arterial system. Therefore, the angiography needs to be visually analysed by a professional in the field of cardiology. Coronary angiography is considered the gold standard for the final diagnosis of arterial diseases through the visual analysis [Ramasamy et al. 2020].

Cardiovascular diseases (CVDs) are a group of disorders that impact the health of the circulatory system and are normally caused by the accumulation of fat cholesterol in the inner walls of arteries in a process called atherosclerosis. According to the World Health Organisation, CVDs are the main causes of death in the world, having affected 17.9 million people in 2019, which represents 32% of global deaths [World Health Organization 2021]. The severity of this problem makes early detection and correct diagnosis two of the most important factors in the treatment of CVDs.

Automatic computer detection or diagnostic systems have risen as a way to assist radiologists in identifying anomalies, providing a second opinion without increasing the workload of the professional [Castellino 2005]. Among the technology systems, machine learning and convolutional neural networks (CNN) stand out as the main ones. CNNs, specially, have been attracting the attention of researchers for their learning capacity added to the automatic extraction of image features.

In a clinical context, diagnostic systems are powered by a digital dataset of images and their efficiency can be weighted by the impact caused on the performance of evaluators. As a rule, the absence of diagnostic systems markings should not eliminate heart lesions identified by the physician, while the presence of unidentified markings should lead to a deeper assessment, therefore, the specialist retains the final word on the diagnosis [Castellino 2005]. The accuracy of detection algorithms or their ability to characterize lesions correctly is linked to the quality of the dataset used for its training and validation.

Jia et al [Jia and Zhuang 2021] made a great research over algorithms for vessel tracking. At their work, several public angiography datasets were presented and what is noticeable is the low quantity of images in each one. This issue was also pointed out by Freitas et al [Freitas et al. 2021]. These public databases provide in the order of dozens of angiographic images, which makes it difficult to develop deep learning algorithms, specially for convolutional neural networks. As an example, studies indicate a minimum of 1500 images per class for classification CNN applications, as well as the requirement for a balanced number of samples between classes [Ponti et al. 2021]. Thus, limited datasets can be the bottleneck for computer-aided diagnosis improvement [Shin et al. 2016]. The limitation of public datasets was also detected in breast radiographic imaging [Dar et al. 2022]. An important direction would be the construction of larger publicly available datasets for medical images [Shen et al. 2017]. Another approach is to train models for automatic data annotation [Kim et al. 2022], lightening the work of specialists and helping to balance the datasets. One explanation to the low quantity of available medical images in datasets may be their approval by ethics committees entities, which is an important and necessary protocol to protect patient privacy.

To mitigate this difficulty, data augmentation is usually used. Data augmentation is an image transformation approach that uses rotations, scale changes and mirroring [Jia and Zhuang 2021] to increase the number of data and is an essential tool to obtain good results in natural images objects recognition CNNs when trained with small datasets. Thus, the data augmentation has its effectiveness verified, increasing the accuracy and sensitivity of CNN algorithms [Lin et al. 2018]. However, if the dataset chosen is poorly represented, the application of data augmentation will also be limited [Ponti et al. 2021] due to the fact that the new images are just modified images from the original dataset.

An alternative to data augmentation is the creation of artificial images through the construction of images that mimic the appearance of vessel elements. Partially synthesised data proved to be useful in improving accuracy in machine learning classifiers [Aljaaf et al. 2016]. The advantage of using artificial images such as those generated by Antczak et al. [Antczak and Liberadzki 2018] for simulating vessels in angiographies is the fact that they are not related to the original dataset. Consequently, the artificial generation of angiographic images provides independent images to be used in training stenosis or aneurysm detection algorithms. Although Antczak's simulation of angiographic ima-

ges was used by other authors presenting good results [Freitas et al. 2022], it was limited to the Bézier function [Ovalle-Magallanes et al. 2022]. Other reports in literature also proposed algorithms for angiogram simulation [Song et al. 2016], and creation of synthetic X-ray images with good results, though they were based on a previously generated 3D model of the organ of interest [Babae and Nilchi 2014] [Kengyelics et al. 2018]. The same approach was used for dental implant recognition [Kohlakala et al. 2022]. The need of several 3D models may interfere with the creation of larger datasets. Another approach was the use of Generative Adversarial Networks (GAN) to generate new X-Ray images of chest [Morís et al. 2021] [Shah et al. 2022] [Srivastav et al. 2021] and stomach [Togo et al. 2019]. GAN algorithms have good results with data augmentation, however the user may lose control of the vessel form in the case of angiographic X-ray images.

Hence, in this work we propose a toolbox for vessel angiography image simulation based on five analytical functions, which allows the generation of datasets with complex vessel structure images while giving the user complete control over the vessel construction, a novelty in literature. Here we also aim to demonstrate how different vessel formats, as well as functions to simulate stenosis and aneurysms, can provide diverse synthetic angiography images by applying random variables to such functions. The toolbox is open source and it is available on GitHub ([https://github.com/jeanschmith/vessel\\_simulator](https://github.com/jeanschmith/vessel_simulator)) as well as an example code and a dataset of right coronary images created with the toolbox.

## 2. Methods

The proposed method for angiography image simulation was based on mathematical equations to draw the vessels. Let an image be represented by a Cartesian's plane in which the width is the  $x$  axis and height is the  $y$  axis. With this definition in mind, we developed five types of vessel trajectories based on equations: line, second degree, sine, Bézier and a special model that we called end of vessel. The points of the plotted equations were approximated to ellipses to manipulate the thickness of the vessels. The user should choose the size of the image in the first place, then select points on the image and choose a vessel trajectory equation to establish its path. After, another set of equations lines can be drawn in that same image to represent a group of vessels. We also developed a function for stenosis and aneurysm simulations. Finally, the user can choose the level of background gradient and white noise in order to simulate images similar to real ones.

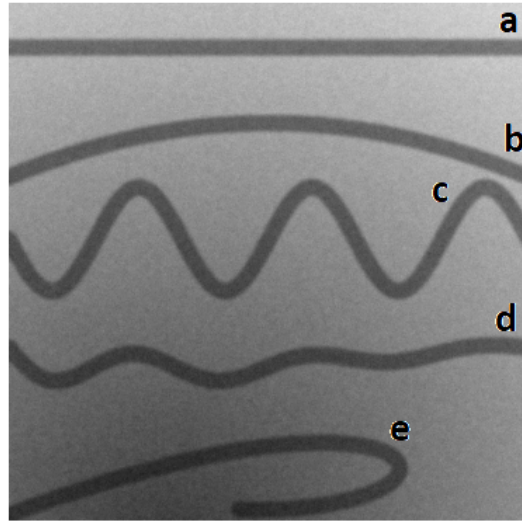
This simulator was developed in Python programming with PIL [Clark 2015], NumPy [Harris et al. 2020] and Matplotlib [Hunter 2007] libraries. The functions of the toolbox are explained in detail in the following subsections.

### 2.1. Line

Although long vessels rarely have the aspect of a straight line, for small segments to connect larger structures this approximation proved to be useful. This is the simplest vessel approximation. The user has two choices for this vessel drawing: providing the start and end points of the line, or the parameters  $a$  and  $b$  for the equation 1.

$$y = ax + b. \tag{1}$$

Where  $y$  represents the height and  $x$  the width of a computed  $P(x, y)$  point in the



**Figura 1. Five types of vessel approximation provided by the toolbox: (a) Line; (b) Second degree; (c) Sine; (d) End of vessel; (e) Bézier.**

image. An example of this type of vessel drawing is presented in Figure 1 with the label *a*.

## 2.2. Second degree

The second degree approach aims to provide a curvature to the vessel. Similarly to equation 1, the second degree function presented in equation 2 can also be calculated using either points or the *c*, *d* and *e* coefficients.

$$y = cx^2 + dx + e. \quad (2)$$

However, providing values for the parable coefficients is not visually as intuitive as it would be for a line. This considered, the user shall provide three  $P(x, y)$  points for the computation of the parameters *c*, *d* and *e* by the algorithm. An example of the second degree type of vessel is shown in Figure 1 twith the label *b*.

## 2.3. Sine

The sine approach intents to add an sinusoidal aspect to the vessel approximating it, for example, to some coronary vessels. The user provides the first and the last point, the amplitude *A* and frequency *f* to draw the vessel. The phase  $\phi$  is optional and the default value is zero. Next, the algorithm computes the width *W* the vessel should occupy in the image by calculating the difference between the start and end *x* points coordinates. The time vector *t* is then computed by making an arrangement between 0 and 1 with the step of  $1/W$ . The amplitude *A* and the *t* vector are applied directly to the equation 3.

$$f_1(t) = A(\sin(2\pi ft + \phi)). \quad (3)$$

Finally, the points computed by the equation 3 are added to a line computed in equation 1, providing an inclination to the vessel in the image. An example of this kind of vessel is presented in Figure 1 with the label *c*.

## 2.4. End of vessel

The aforementioned equations are good approximations to vessels structures and connection between them. However there are cases when the vessel presents more than one function to determine the trajectory behaviour. Therefore, we decided to implement a function called *end of vessel*. This formulation treats the cases when the vessel is attached to a tissue and is not connected to other vessels in both extremities.

To use this function, the user provides the first and the last point to draw the vessel with an amplitude  $A$ . Next, the algorithm computes the width  $W$  and the time vector  $t$  the same way it was computed for the sine function as previously mentioned. The amplitude  $A$  and the  $t$  vector are applied to equation 4.

$$f_2(t) = A \left( \sin(2\pi ft) + 2\sin\left(2\pi\frac{1}{2}t\right) \right) e^{-\alpha t}. \quad (4)$$

One might note that the equation 4 is a sum of two different sine signals with a relation of double the amplitude and half the frequency. This formulation gives an aspect in the vessel drawing more similar to variations in real images. Finally, a decay is applied to the sinusoidal functions to decrease the amplitude of signal smoothly. The default value of the decay is  $\alpha = -2$  while the frequency  $f$  was defined as a random number between 2 and 4 hertz. These parameters can be altered by the user at any time. Finally, the points computed by equation 4 are added to a line computed in equation 1 to give the slope of the vessel in the image. The vessel labeled  $d$  in Figure 1 exemplifies an application of the *end of vessel* function.

## 2.5. Bézier

The Bézier curve used was the same function as the one in the Antczak et al. [Antczak and Liberadzki 2018] work. They designed a line in the image based on the third order of Bézier lines, the cubic Bézier curve, presented in equation 5.

$$B(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3. \quad (5)$$

The image created with this equation is a polygon that connects its four control points, starting and ending at the first and last point and whose shape is determined by the position of the control points remaining, without necessarily passing through these points. In this case, the control points  $P_0(x, y)$ ,  $P_1(x, y)$ ,  $P_2(x, y)$  and  $P_3(x, y)$  should be provided by the user and the vector  $t$  is automatically computed by the algorithm. An example is presented in Figure 1 by the vessel labeled  $e$ .

## 2.6. Normal, stenosis and aneurysm

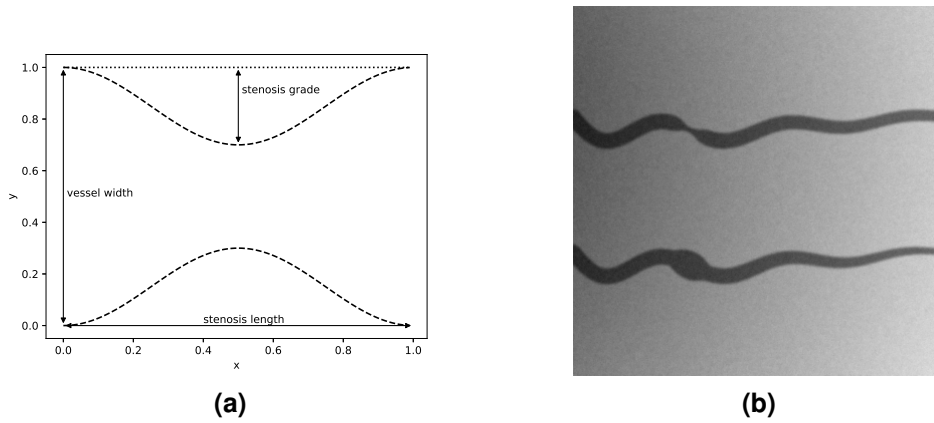
Once the vessel trajectory equation is computed, these points shall pass through a process to give the vessel width  $V_w$ . For the *normal* situation, each point is transformed into an ellipse. We defined a bounding box where the ellipses are drawn and its definition for each point  $P(x, y)$  is  $P_a(x - V_w, y - V_w)$  and  $P_b(x + V_w, y + V_w)$ . Furthermore, an array of bounding boxes for the ellipses drawing process using PIL is generated. All vessels in Figure 1 are presented with the *normal* width process which means they have no stenosis or aneurysms.

The developed toolbox provides an option to include stenosis in the vessels. From the points computed by equations 1, 2, 3, 4 and 5, the user shall provide the position, the grade  $S_g$  and length  $S_l$  of the stenosis in the vessel. In this function the bounding boxes are also computed to draw the ellipses, but their sizes were in respect to the stenosis equation 6.

$$S_f(t) = \cos\left(2\pi\frac{1}{S_l}t\right) \quad (6)$$

The time vector  $t$  is computed to envelop exactly one period of the cosine function with respect to  $S_l$ , and the frequency is  $1/S_l$ . For a better understanding of the stenosis equation, please refer to Figure 2a for a normalised case. We defined that the stenosis should follow the form of a cosine. After the computation of the equation 6 the output values of  $S_f(t)$  are mapped from the limits of  $(-1, 1)$  to  $(1 - S_g, 1)$ . Next, the mapped version of  $S_f(t)$  is multiplied by  $V_w$  to get the stenosis effect on the original height of the vessel. Note that this process will produce the ellipses bounding boxes with respect to equation 6 form. This process occurs only at the position provided by the user until the stenosis length  $S_l$ . The *normal* case is applied to the rest of the vessel.

The same process applied to stenosis was used in the case of aneurysm simulation. The only difference was in the mapping of the limits, adjusting it to  $(-1, 1)$  to  $(1 + S_g, 1)$ . An example of two vessels with stenosis and aneurysm respectively are illustrated in Figure 2b.



**Figure 2. Stenosis schematic using equation 6 (a). Examples of stenosis (upper vessel) and aneurysm (lower vessel) simulations. In those vessels it was also applied a thickness manipulation, meaning in the left side the vessels begin with a certain width that decreases as the flow reaches the right side of the image (b).**

## 2.7. Thick manipulation

Vessels located at extremities of larger structures, specially those attached to tissues, have their thickness altered so by the end they appear thinner. In this case, to get these simulations closer to what would be expected in a real vessel, the *thick manipulation* was developed.

The *thick manipulation* is done using the vector of ellipses already computed in the normal, stenosis or aneurysm functions and a thickness factor provided by the user. This factor can vary from 0 to 1 according to the position in which the equation 7 is going to start being applied. The change in factor gives the possibility of choosing the desired thickness of the end vessel, with factor 1 meaning that the whole vessel will have its ellipses size modified and, consequently, factor 0 means the vessel won't receive any change.

$$T(t) = e^{\alpha t}. \quad (7)$$

Here, the decay  $\alpha$  received a default value of 1 and  $t$  representing the length of the vessel was computed by the algorithm. This function results in a new definition for the bounding boxes for each point  $P(x, y)$  and the radius of the ellipses are decreased with respect to equation 7. An example of thick manipulation is presented in Figure 2b.

## 2.8. Gradient

The gradient function was also the same as presented by the Antczak et al. [Antczak and Liberadzki 2018] work, in which the function is responsible for the gray scale background of the image. First, the background has its centre position defined randomly based on the image's size. Looping through all the pixels in the image, a distance to the centre is calculated for every pixel through equation 8, where  $x_c$  and  $y_c$  are the center's coordinates.

$$d(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2}. \quad (8)$$

Finally, a colour is assigned to each pixel taking into account the distance. The pixels closer to the centre will receive darker shades of gray while the ones farther away from the centre will be lighter.

## 2.9. White noise

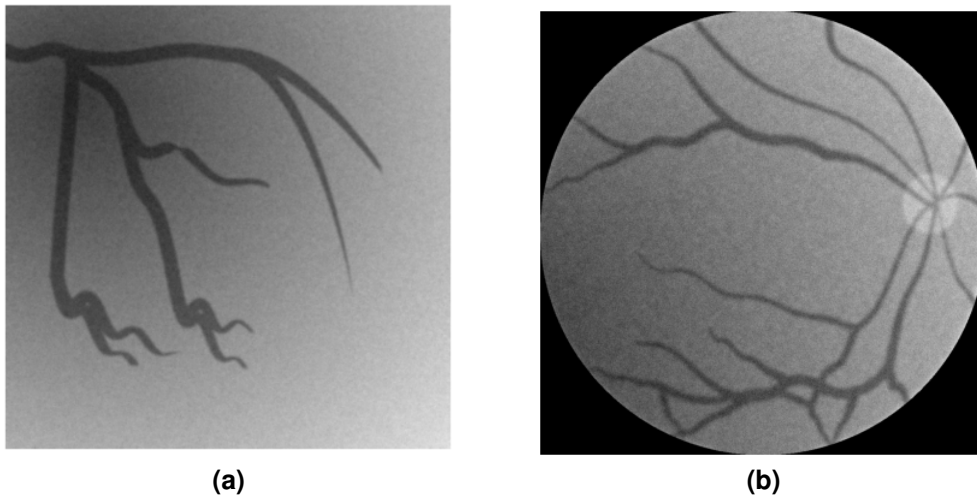
By the end of the process, white noise is added to the images to create imperfections closer to the noise presented in real ones. Similar to the previous section, all the pixels had their values changed. A variable was randomly generated ranging from  $-\delta$  to  $\delta$ , dependant on the image's height to represent the white noise for each pixel. The noise grade is defined by user from 0 to 1. The white noise with respect to the defined grade is added to each pixel and latter a Gaussian blur is applied to the image.

## 3. Results

The toolbox provided good results in simulating complex image structures. As an example, in Figure 3a is presented a left coronary angiography image simulation. To construct this image all the trajectory functions were used. Furthermore, an stenosis can be observed in one of the vessels. One might note the gradient effect in the background and the white noise addition.

As an second example, in Figure 3b is presented a retinal vessel structure to simulate a fluorescein angiography image. Again, to simulate this complex vessel structure all

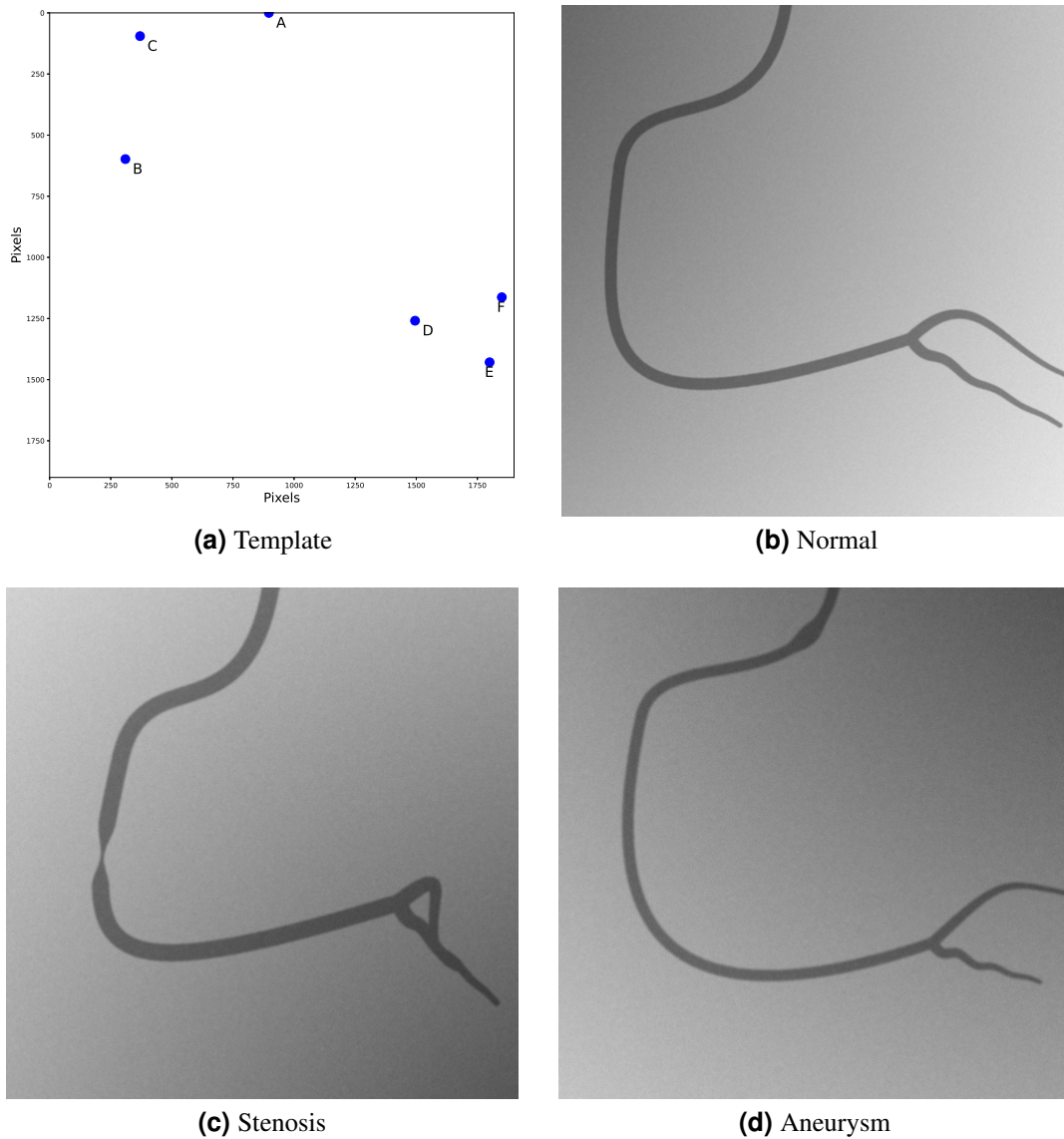
functions of the toolbox were used. Naturally, the number of vessels and the complexity of the image is up to the user. After the construction of the vessel structure, the resulting image was additionally modified, in this case, a round border was added with a simple mask. Moreover, a round and lighter mask was added to represent the optic nerve region. One might also observe that this image has more noise than the left coronary one.



**Figura 3. (a) Example of left coronary angiography with stenosis image simulation. In this example all trajectory functions presented in this work were used. This example also presents thick manipulation, gradient background and white noise applications. (b) Example of a retinal vessel structure to simulate a fluorescein angiography image. All functions of the toolbox were used. Further image transformations were applied after the simulation by the toolbox was complete, like the round border (using a simple mask) and the round lighter mask used to represent the optic nerve region in the image.**

As aforementioned, the toolbox is open source and available on GitHub, where we also provided a right coronary image dataset constructed with the toolbox. The dataset is composed by a hundred images of normal right coronary, a hundred images with varying degrees of stenosis in the right coronary and a hundred images with varying degrees of aneurysms in the right coronary. In Figure 4a a template to start drawing vessels was established. Between the points A, B and C the Bézier function was used. Between the points B to D, and D to F, again the Bézier function was used. Finally, between the points D and E the end of vessel was used. All the points from the template had slightly random variations in their positions to create different structures. The parameters were chosen randomly with respect to configured limits, as well. The vessel positions, based on the points A,B,C,D and E, had a limit in the random choice to keep the right coronary aspect, the vessel width was limited from 20 to 35, the position of stenosis or aneurysm was limited to be placed from the beginning until the bifurcation, the grade was limited from 0.4 to 0.8 and the length was limited from 40 to 80. Figure 4b presents an example of a normal right coronary extracted from the dataset. Figure 4c presents an example of a right coronary with stenosis extracted from the dataset. The configured stenosis grade was 0.8 and length of 49. Figure 4d presents an example of a right coronary with aneurysm. The configured aneurysm grade was 0.7 and length of 54.





**Figura 4.** Examples of images from the dataset created with the toolbox. The images have the size of 1920x1920 pixels. (a) A template to start drawing different vessel structures was defined. A normal right coronary (a), a stenosis with grade of 0.8 and length of 49 was added to the right coronary (b) and an aneurysm with grade of 0.7 and length of 54 was added to the right coronary (c).

## 4. Discussion

As presented in literature, simple Bézier functions to provide synthetic stenosis vessel images proved to increase the results in CNN implementations [Freitas et al. 2022] [Ovalle-Magallanes et al. 2022]. Therefore, we have the conviction that the images simulated by the toolbox presented in this work are suitable for learning algorithms training and help to fill the lack of natural images problem. Moreover, the toolbox can provide images that the Bézier approximations have difficulties making (bifurcations, for example), besides being able to help train more general CNNs.

Although the examples presented in Figure 3a and 3b are clear simulations of the real images, this is the closer approximations to real images presented in the literature for the best knowledge of the authors since other studies used only the Bézier functions [Freitas et al. 2022] [Ovalle-Magallanes et al. 2022] [Antczak and Liberadzki 2018]. The novelty is in having more than one trajectory analytical function and the potential to simulate complex structures of angiography images. Also, the severity of stenosis and aneurysms in synthetic vessel images is up to user. Therefore, the presented toolbox provide the possibility of any angiographic image simulation.

In addition, in this discussion we aimed to provide an alternative to the lack of datasets for leaning algorithms, but another usage of simulated X-ray images is for academic purposes [Kengyelics et al. 2018]. The real X-ray angiographic images might be rare, and therefore, any specific situation can be simulated for educational improvements, or even, to emulate some particular cardiovascular disease.

The advantage of using the proposed toolbox is that there is no need of training processes like in GAN methods. Moreover, GAN methods demand previous real images for training [Morís et al. 2021] [Shah et al. 2022][Srivastav et al. 2021] [Togo et al. 2019], but in the case of angiographic X-ray the freely available images are scarce and the user may also loose control of the vessel form in this specific case.

3D methods produced excellent results [Song et al. 2016], though the model should be previously computed. For dataset building, creating several 3D models may be a laborious task and time consuming. Naturally, the toolbox needs improvements. In the future, new trajectory equations might be added to the toolbox portfolio.

## 5. Conclusion

In this work we presented an alternative to traditional data augmentation methods, consisting in a toolbox for X-ray angiography image simulations through vessel trajectory drawing with stenosis and aneurysms mimicking functions. The variability and realness of synthetic images are provided not only by the addition of random variables, but also through background gradient and white noise functions. Simulated images have been proved useful for building datasets or balancing data classes for any computational algorithm implementation that suffer from the low availability of real medical images, such as CNN learning algorithms, which may require thousands of images for accurate training. Future works include the evaluation of a CNN model performance when trained with our synthetic data. The toolbox is open source and available on GitHub along with an example code and a dataset of right coronary images created with the toolbox. Improvement contributions are most welcome.

## Referências

- Aljaaf, A. J., Al-Jumeily, D., Hussain, A. J., Fergus, P., Al-Jumaily, M., and Hamdan, H. (2016). Partially synthesised dataset to improve prediction accuracy. In *International Conference on Intelligent Computing*, pages 855–866. Springer.
- Antczak, K. and Liberadzki, Ł. (2018). Stenosis detection with deep convolutional neural networks. In *MATEC Web of Conferences*, volume 210, page 04001. EDP Sciences.

- Babae, M. and Nilchi, A. R. N. (2014). Synthetic data generation for x-ray imaging. In *2014 21th Iranian Conference on Biomedical Engineering (ICBME)*, pages 190–194. IEEE.
- Castellino, R. A. (2005). Computer aided detection (cad): an overview. *Cancer Imaging*, 5(1):17.
- Clark, A. (2015). Pillow (pil fork) documentation. *readthedocs*.
- Dar, R. A., Rasool, M., Assad, A., et al. (2022). Breast cancer detection using deep learning: datasets, methods, and challenges ahead. *Computers in Biology and Medicine*, page 106073.
- Freitas, S. A., Nienow, D., da Costa, C. A., and Ramos, G. d. O. (2021). Functional coronary artery assessment: a systematic literature review. *Wiener klinische Wochenschrift*, pages 1–17.
- Freitas, S. A., Zeiser, F. A., Da Costa, C. A., and Ramos, G. D. O. (2022). Deepcadd: a deep learning architecture for automatic detection of coronary artery disease. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. (2020). Array programming with numpy. *Nature*, 585(7825):357–362.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95.
- Jia, D. and Zhuang, X. (2021). Learning-based algorithms for vessel tracking: A review. *Computerized Medical Imaging and Graphics*, page 101840.
- Kengyelics, S. M., Treadgold, L. A., and Davies, A. G. (2018). X-ray system simulation software tools for radiology and radiography education. *Computers in biology and medicine*, 93:175–183.
- Kim, D., Chung, J., Choi, J., Succi, M. D., Conklin, J., Longo, M. G. F., Ackman, J. B., Little, B. P., Petranovic, M., Kalra, M. K., et al. (2022). Accurate auto-labeling of chest x-ray images based on quantitative similarity to an explainable ai model. *Nature communications*, 13(1):1–15.
- Kohlakala, A., Coetzer, J., Bertels, J., and Vandermeulen, D. (2022). Deep learning-based dental implant recognition using synthetic x-ray images. *Medical & Biological Engineering & Computing*, 60(10):2951–2968.
- Lin, Y., Zhang, H., and Hu, G. (2018). Automatic retinal vessel segmentation via deeply supervised and smoothly regularized network. *IEEE Access*, 7:57717–57724.
- Morís, D. I., de Moura, J., Novo, J., and Ortega, M. (2021). Portable chest x-ray synthetic image generation for the covid-19 screening. *Engineering Proceedings*, 7(1):6.
- Ovalle-Magallanes, E., Avina-Cervantes, J. G., Cruz-Aceves, I., and Ruiz-Pinales, J. (2022). Improving convolutional neural network learning based on a hierarchical bezier generative model for stenosis detection in x-ray images. *Computer Methods and Programs in Biomedicine*, 219:106767.

- Ponti, M. A., dos Santos, F. P., Ribeiro, L. S., and Cavallari, G. B. (2021). Training deep networks from zero to hero: avoiding pitfalls and going beyond. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 9–16. IEEE.
- Ramasamy, A., Chen, Y., Zanchin, T., Jones, D. A., Rathod, K., Jin, C., Onuma, Y., Zhang, Y.-J., Amersey, R., Westwood, M., et al. (2020). Optical coherence tomography enables more accurate detection of functionally significant intermediate non-left main coronary artery stenoses than intravascular ultrasound: a meta-analysis of 6919 patients and 7537 lesions. *International Journal of Cardiology*, 301:226–234.
- Shah, P. M., Ullah, H., Ullah, R., Shah, D., Wang, Y., Islam, S. u., Gani, A., and Rodrigues, J. J. (2022). Dc-gan-based synthetic x-ray images augmentation for increasing the performance of efficientnet for covid-19 detection. *Expert Systems*, 39(3):e12823.
- Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221.
- Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., and Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298.
- Song, S., Yang, J., Fan, J., Cong, W., Ai, D., Zhao, Y., and Wang, Y. (2016). Geometrical force constraint method for vessel and x-ray angiogram simulation. *Journal of X-ray Science and Technology*, 24(1):87–106.
- Srivastav, D., Bajpai, A., and Srivastava, P. (2021). Improved classification for pneumonia detection using transfer learning with gan based synthetic image augmentation. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 433–437. IEEE.
- Togo, R., Ogawa, T., and Haseyama, M. (2019). Synthetic gastritis image generation via loss function-based conditional pggan. *IEEE access*, 7:87448–87457.
- Vijayalakshmi, I. (2015). *Cardiac Catheterization and Imaging (From Pediatrics to Geriatrics)*. JP Medical Ltd.
- World Health Organization (2021). Cardiovascular disease (cvds). [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).