UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

telecos
BCN

ČVUT
ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

# Processing in vivo ultrasound images of the carotid artery

Degree Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Ana Manzano Rodríguez

In partial fulfillment
of the requirements for the degree in

*Telecommunications Technologies and Services* **ENGINEERING**

Tutor: Jan Kybic
Supervisor: Veronica Vilaplana

Prague, June 2021

This Thesis has been done in the Department of Cybernetics of the Faculty of Electrical Engineering of the **Czech Technical University** in Prague

# Abstract

Carotid stenosis is a narrowing of the carotid arteries, the two major arteries that carry oxygen-rich blood from the heart to the brain. This disease is caused by a buildup of plaque (atherosclerosis) inside the artery wall that reduces blood flow to the brain. This thesis focuses on predicting whether the plaque in the carotid artery is unstable (symptomatic) or stable (asymptomatic) using ultrasound images of the carotid artery. If it is unstable it means that the plaque is going to grow, otherwise, is going to remain the same. Using a provided segmentation, a series of descriptors and a subsequent classification model has been developed to fulfil this task.

We will see that between the linear regression classifier, SVC or Random Forest, SVC will give the best results. For the cross-sectional images, the descriptors that will give us the best accuracy in distinguishing the two classes will be: relative percentage stenosis, relative plaque area, wavelets and Haralick texture descriptors. The first two will be calculated on the cross-sectional segmentations and the last ones on the original cross-sectional images using segmentations as well. With this selection of features we will achieve 67% accuracy in the classification of our data.

# Resum

L'estenosi carotídia és un estrenyiment de les artèries caròtides, les dues artèries principals que porten la sang rica en oxigen del cor al cervell. Aquesta malaltia està causada per una acumulació de placa (aterosclerosi) a l'interior de la paret arterial que redueix el flux sanguini al cervell. La tesis que es presenta es centra en predir si la placa en l'arteria caròtida es inestable (simptomàtica) o estable (asimptomàtica) utilitzant les imatges ecogràfiques de l'arteria caròtida. Si és inestable significa que la placa creixerà, d'altra banda, si és estable, romandrà igual. Mitjançant una segmentació que se'ns ha facilitat, s'han desenvolupat una sèrie de descriptors i un posterior model de classificació per complir aquesta comesa.

Veurem que entre el classificador de regressió lineal, SVC o Random Forest, SVC serà amb el que obtindrem millors resultats. Per les imatges transversals, els descriptors que ens donaran una major precisió al distingir les dos classes seran: percentatge d'estenosis relativa, àrea relativa de la placa, wavelets i els descriptors de textura de Haralick. Les dues primeres es calcularan sobre les segmentacions transversals i les últimes sobre les imatges transversals originals utilitzant també les segmentacions. Amb aquesta selecció de característiques s'aconseguirà un 67% de precisió en la classificació de les nostres dades.

# Resumen

La estenosis carotídea es un estrechamiento de las arterias carótidas, las dos arterias principales que llevan la sangre rica en oxígeno del corazón al cerebro. Esta enfermedad está causada por una acumulación de placa (aterosclerosis) en el interior de la pared arterial que reduce el flujo sanguíneo al cerebro. La presente tesis se centra en predecir si la placa en la arteria carótida es inestable (sintomática) o estable (asintomática) utilizando las imágenes ecográficas de la arteria carótida. Si es inestable significa que la placa va a crecer, por otra parte, si es estable, se mantendrá igual. Mediante una segmentación que nos ha sido facilitada, se han desarrollado una serie de descriptores y un posterior modelo de clasificación para cumplir este cometido.

Veremos que entre el clasificador de regresión lineal, SVC o Random Forest, SVC será el que nos dará mejores resultados. Para las imágenes transversales, los descriptores que nos darán una mayor precisión al distinguir las dos clases serán: porcentaje de estenosis relativa, área relativa de la placa, wavelets y los descriptores de textura de Haralick. Las dos primeras se calcularán sobre les segmentaciones transversales y las últimas sobre las imágenes transversales originales utilizando también las segmentaciones. Con esta selección de características se conseguirá un 67% de precisión en la clasificación de nuestros datos.

# Acknowledgements

I would like to thank Professor Jan Kybic for allowing me to work on this project, guiding me through it and offering his help whenever I needed it.

Dedicated to my family and friends, to those in Barcelona and the new ones in Prague, who have been supporting and encouraging me all along the way.

# Contents

# List of Figures

# List of Tables

# List of Acronyms and Abbreviations

**AS** Aortic Stenosis. 7

**CAS** Carotid Artery Stenosis. 10, 12, 13

**CCA** Common Carotid Artery. 11

**CTA** Computed Tomography Angiography. 13

**DWF** Discrete Wavelet Frame. 23

**ECA** External Carotid Artery. 11

**GLCM** Gray-Level Co-occurrence Matrix. 21, 22

**ICA** Internal Carotid Artery. 11

**MRA** Magnetic Resonance Angiography. 13

**SVC** Support Vector Classification. 30

**SVM** Support Vector Machine. 30

**TIA** Transient Ischemic Attack. 13

# Chapter 1

# Introduction

Aortic stenosis (AS) is the most common valvular heart disease and the third most common cardiovascular disease. This illness restricts the blood flow from the left ventricle to the aorta and may also affect the pressure in the left atrium. It can cause the patient to feel weak, faint, or lethargic and can even lead to heart failure [3].

The prevalence of AS increases from 2% in adults over 65 years to 4% in adults over 85 years of age and the problem is expected to grow in the decades to come as the mean lifetime expands. It is a progressive condition, meaning that it gets worst over time. After the onset of heart failure, survival is $< 2$ years without valve replacement [25]. That is why it is very important to be able to detect it early in order to improve the quality of life and prevent more serious consequences.

A very novel strategy that is starting to be implemented in the detection of AS is to use machine learning techniques to automatically predict whether or not the disease might be developing. Machine learning, which is a subfield of Artificial intelligence, is the study of computer algorithms that improve automatically through experience and by the use of data or information [17]. It is applied in many fields, among them agriculture, finance, audio, image and video processing and medical diagnostics can be found.

The aim of this project is to predict whether the plaque in the carotid artery is unstable (symptomatic) or stable (asymptomatic) using ultrasound carotid artery images. The analysis of this images is going to be applied for a long term development prediction. Given a prior segmentation which will be provided by Martin Kostelanský [18] (e.g. Figure 3.1), the objective is to implement descriptors (e.g. size and shape of the plaque), and then try to predict the outcome from the descriptors.

## 1.1 Goals

There are four main goals for this project.

1. Firstly, we will do an analysis of the recently research.
2. Secondly, we will implement the descriptors in Python using different image processing techniques, as well as mathematical functions.
3. Thirdly, we will implement the classification, using different classifiers.
4. Lastly, the goal will be the experimental and statistical evaluation of the performance. With the results obtained, we will try to improve the model.

## 1.2 Project development

The project was carried out at Czech Technical University in Prague. It has been developed in the Department of Cybernetics of the Faculty of Electrical Engineering which is a research and teaching unit in the fields of artificial intelligence, machine perception, cybernetics, robotics, and biomedical engineering with the focus on computer vision, machine learning, pattern recognition, knowledge-based systems, medical data processing, autonomous and collaborative robotics.

This project is a continuation of a thesis that was carried out at the same university by Martin Kostelanský called "Localization and segmentation of in-vivo ultrasound carotid artery images" [18]. In it, three separate image recognition tasks were treated- classification, localization and segmentation of the ultrasound images of the carotid artery with stenosis. In classification, the images were categorized into four classes of the ultrasound images (longitudinal, transverse, Doppler, conical). The region of interest, the carotid artery, was localized on the transverse and longitudinal images. The area of an artery was segmented into artery wall with plaque, lumen, and surrounding tissue.

In this project a more precise segmentation, provided by Martin Kostelanský, will be used posterior the above-mentioned thesis [18]. We will work with the transverse and longitudinal segmentations. We can find an example of this images in Figure 1.1.

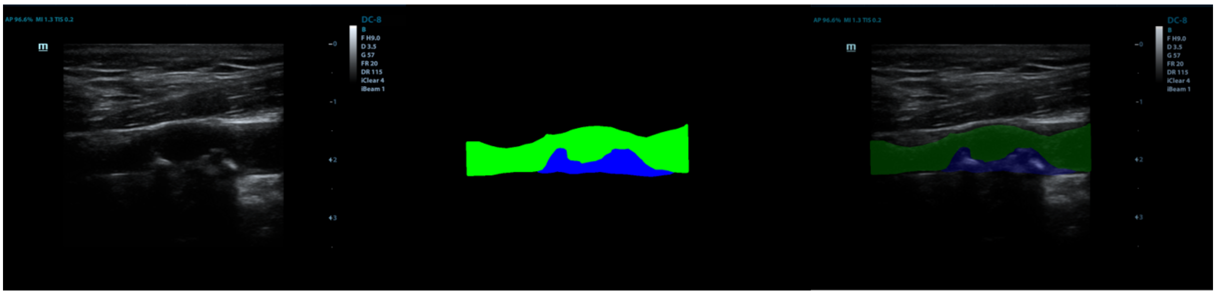


Figure 1.1: At left, we can see the original ultrasound carotid artery image, in the centre, the segmented image and on the right the segmentation overlaid over the original image

The details of the implementation of this project can be found at Appendix F.

## 1.3 Budget

In this section we will deal with the total cost of project development. In order to carry it out, the main expense is the student who develops it and the teachers/supervisors from both universities.

In our case, one supervisor of the CTU and one tutor from UPC were involved. To fulfill the 18 credits of the thesis, is expected to dedicate about 25 hours/credit, thus approximating a total of 450 hours. The CTU supervisor attended, approximately in total, 1h weekly meeting for 18 weeks. Counting extra work for the evaluation, about 23 hours of work are considered to be directly dedicated to this project. With the UPC tutor, as there is no continuous weekly evaluation, approximately 15 hours of work are estimated, taking into account coordination and evaluation. The total budget is 6.509€ as we can see in Table 1.1.

| | Number | Salary(€)/hour | Hours | Salary(€) | Taxes 21% | Total(€) |
|---|---|---|---|---|---|---|
| **Student** | 1 | 9 | 495 | 4.050 | 850 | 4900 |
| **Supervisor CTU** | 1 | 35 | 23 | 805 | 169 | 974 |
| **Tutor UPC** | 1 | 35 | 15 | 525 | 110 | 635 |
| **BUDGET** | | | | | | **6.509** |

Table 1.1: Budget related to the staff and student

For the materials part, we must take into account the computer, the server where the data is stored and the office, as the student was offered a space in the department to be able to work on the project. Considering the computer of the student is 1.300€, with a life expectancy of 7 years (84 months) and a residual value of the 25%. We can calculate the amortizations as:

$$f = \frac{initial\_total - residual\_value}{life\_expectancy}$$

For the server we can approximate the cost to 1.025€, with a life of 120 months and a residual value of a 10%. Finally, for a space in the office we would pay 100€ for 5 months, with 70 years of life and a residual value of 5%. The cost of the materials and space is as show in Table 1.2, where the use of them is of 5 months.

| | **Cost(€)** | **Amount** | **Initial total(€)** | **Residual value(€)** | **Amortization(€)** | **Total(€)** |
|---|---|---|---|---|---|---|
| **Computer** | 1.500 | 1 | 1.500 | 375 | 14 | 70 |
| **Server** | 1.025 | 1 | 1.025 | 102 | 8 | 40 |
| **A space in the office** | 100 | 1 | 100 | 5 | 0,12 | 0,6 |

Table 1.2: Total amortization related to materials and space

## 1.4 Structure of the document

This final degree thesis report is structured as follows: Chapter 1 provides a brief introduction of the aim of the thesis, explains the context of the project and how it will be structured. In Chapter 2 we can find the technical background about CAS as well as the techniques to be applied on image processing.

Chapter 3 will describe the development of the project starting from the original dataset. It will explain all the descriptors used and how these results have been applied to obtain a machine learning model that is able to classify our data.

In Chapter 4 the results are going to be described and analysed through histograms and statistics. The conclusions of this thesis can be found in Chapter 5, giving a brief summary of what has been achieved with the work done.

# Chapter 2

# Background

In this chapter the state of art of the project is explained. Starting with a brief explanation of what the carotid arteries are in Section 2.1 and what is the disease that concerns us in this project, the carotid artery stenosis, in Section 2.2. The diagnosis and evaluation of the illness as well as the treatment are going to be explained in the subsections of this section.

In the second half of the chapter we will focus on the image processing techniques undertaken, that will be detailed in Section 2.3. We will use them to get the information we need from our images.

## 2.1 The carotid arteries

The carotid arteries are major blood vessels in the neck that supply blood to the neck, face, and brain. There are two carotid arteries, one on the right and one on the left that are a bifurcation of the common carotid artery (CCA). In the neck (at the level of the C4 - C5 intervertebral space [12]), each carotid artery branches into two divisions:

- The internal carotid artery (ICA) supplies blood to the brain.
- The external carotid artery (ECA) supplies blood to the face and neck.

Like all arteries, the carotid arteries are made of three layers of tissue: intima, which is the smooth innermost layer, the media, which is the muscular middle one and adventitia, the outer layer. The carotid sinus, or carotid bulb, is a widening of a carotid artery at its main branch point. It contains sensors that help regulate blood pressure [6].

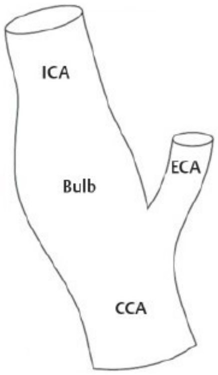


Figure 2.1: Carotid arteries

## 2.2 Carotid artery stenosis

Carotid artery stenosis (CAS) is a disease caused by the narrowing in the large arteries located on each side of the neck that carry blood to the head, face and brain. The narrowing usually results from atherosclerosis*, or a build-up of plaque on the inside of the arteries as can be seen in Figure 2.2. Eventually, stenosis can advance to complete obstruction of the artery.

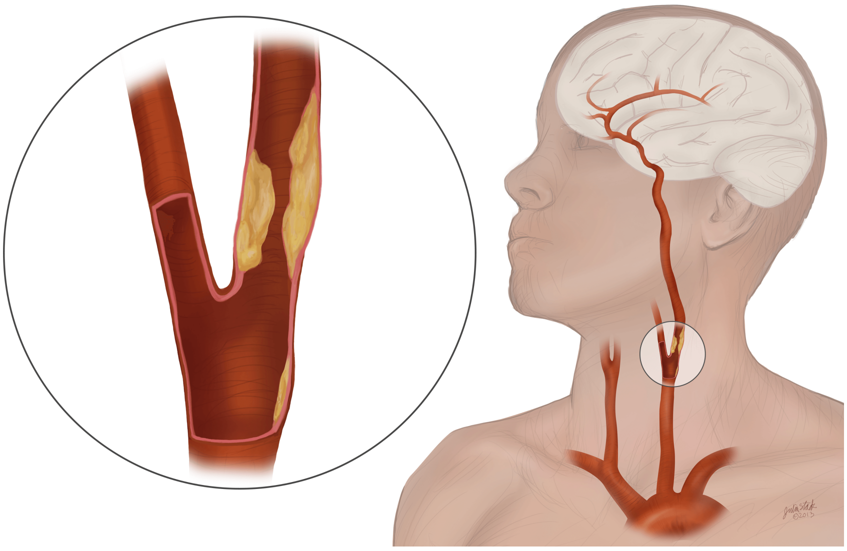


Figure 2.2: Carotid artery stenosis

* Atherosclerosis is a potentially serious condition where arteries become clogged with fatty substances called plaques, or atheroma [4].

Risk factors for carotid artery stenosis include high blood pressure, age, smoking, obesity, diabetes and an inactive lifestyle.

Some people with CAS may experience fainting, dizziness and blurred vision which may be signs of the brain not receiving enough blood. In many cases, the first symptom is a transient ischemic attack (TIA) or a stroke because a small blood clot can form in the area of the vessel that is affected by atherosclerosis. When such a small clot becomes dislodged, it can travel into the brain and plug up a smaller artery on which a particular piece of the brain depends for its function and ultimately survival. Symptoms of a TIA and stroke are similar: paralysis or numbness on one side of the body, blurred vision, headache, trouble speaking and difficulty responding to others. A TIA is usually brief and leaves no lasting damage; it is due to a very small, temporary occlusion of a small artery but often a warning sign. A stroke is often associated with permanent injury of a part of the brain due to loss of its blood supply and can result in severe disability or death.

### 2.2.1 Diagnosis and evaluation

CAS sometimes causes an abnormal sound, or bruit, in the artery that can be heard with a stethoscope*. Imaging tests to diagnose, localize and measure stenosis include:

- **Carotid ultrasound** (including Doppler ultrasound): This test uses sound waves to create real-time pictures of the arteries and locate blockages. Doppler is a special ultrasound technique that can detect areas of restricted blood flow in the artery [30].
- **Computed tomography angiography** (CTA): CTA uses a CT scanner to produce detailed views of the arteries anywhere in the body – in this case, in the neck [22].
- **Magnetic resonance angiography** (MRA): This noninvasive test gives information similar to that of CTA without using ionizing radiation [9].
- **Cerebral angiography:** Is a minimally invasive test in which a catheter is guided through an artery to the area of interest. Contrast material is injected through the tube and images are captured with x-rays [11].

### 2.2.2 Treatment

Severe cases of stenosis often require carotid endarterectomy, in which a surgeon makes an incision to remove plaque and any diseased portion of the artery while the patient is under general anesthesia. A less invasive option would be the carotid artery angioplasty and stenting. During this procedure, a catheter is threaded from an incision in the groin

* Acoustic medical device for auscultation, or listening to internal sounds of an animal or human body [27]

to the site of the blockage, where a balloon tip is inflated to open the artery as we can see in Figure 2.3. A stent may be placed in the artery to expand it and hold it open [10].

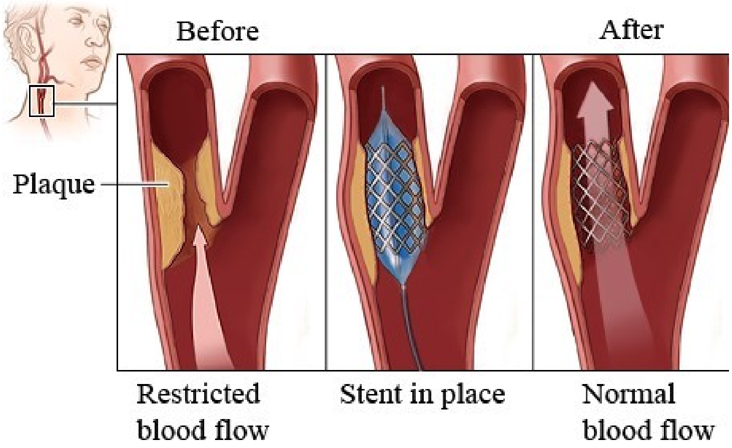


Figure 2.3: Carotid artery angioplasty and stentings

## 2.3 Image processing techniques

Image preprocessing is one of the most important parts when creating a machine learning model. It involves using various techniques to extract the relevant information for our proposition. For example, if we only want to work with the plaque class and obtain information from it, we will only use the pixels that have been defined as plaque. In our case, we are using a segmentation that it is already given by Martin Kostelanský [18]. Segmentation is a technique that consists on the partition of an image into multiple parts or regions, often based on the characteristics of the pixels in the image. This is going to be very useful to distinguish the different parts of the carotid artery and work with them separately with the different descriptors.

Another technique that will be very useful in the analysis of the segmentations will be the masking. A mask, which is a binary image consisting of zero and non-zero values, is defined and it would be used as a parameter together with another binary or grayscale image to the operation in question. The operator (e.g. texture descriptor) needs to know what pixels to consider. All pixels which are zero in the mask will be understood as ”outside” in the output image so the calculations will not involve them if desired. This will be useful when we want to make calculations with only a part of the carotid artery image.

# Chapter 3

# Methodology

This chapter aims to explain the work conducted. In order to predict if the plaque of the image is stable or unstable, we will have to treat and prepare the dataset beforehand, as explained in Section 3.1. The descriptors will then be implemented for further classification. This will be spelled out in more detail in Section 3.2. The output of the descriptors will give us the necessary tools to create a discrete database with which to train our model and obtain predictions. This process is explained in Section 3.3.

The chapter ends with an explanation of the model implementation. We are going to describe the different classifiers that are being used to classify our data in Section 3.3.3. Finally, we are going to explain how to improve the performance with feature selection in Section 3.3.4.

## 3.1 Dataset

A dataset in machine learning [15] is an indexed set of $N$ pairs of observations $x_i$ and labels $y_i$. The observations can be vectors but also images or other type of data.

In our project, we have a document called *data_description.csv* (that has been provided with the segmentation, see Appendix B) in which we can find all the images of our dataset labelled. In this document, there are 14003 patient ultrasound carotid artery images labelled as *progressive* and 11830 ultrasound carotid artery images labelled as *stable*.

For a subset of the images we have an expert segmentation that has 149 longitudinal and 151 cross-sectional segmented images that are part of the initial dataset of 25833 ultrasound carotid artery images. This 300 images have been labelled manually by experts. We can see two examples in Figure 3.2 and Figure 3.1. For the rest of images, the idea is to have automatic segmentations later on.

In Figure 3.1 we can see an example of transversal segmentation. It consists of three distinct parts: the vessel wall (surrounding tissue), which is formed by the red pixels, the lumen*, formed by green pixels, and finally the plaque, represented by the blue ones.

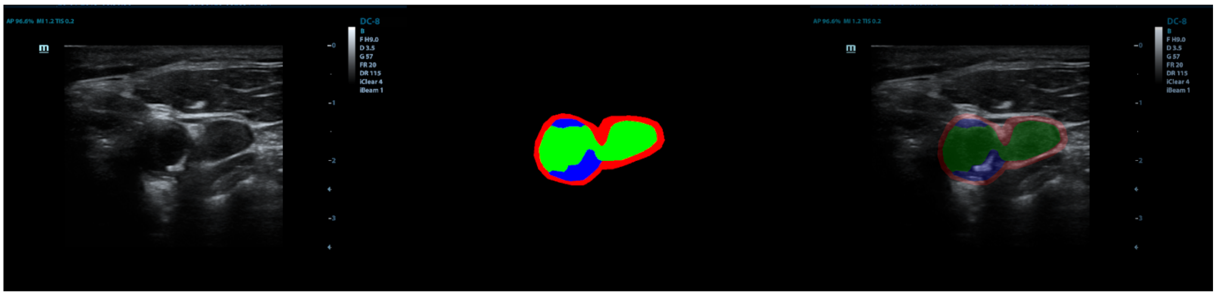


Figure 3.1: At left, we can see the original ultrasound carotid artery image, in the centre, the segmented image and on the right the segmentation overlaid over the original image

We now look at the longitudinal segmentation. The experts decided that they could not distinguish the wall in all images, so they decide it not to mark it. We can see an example in Figure 3.2.

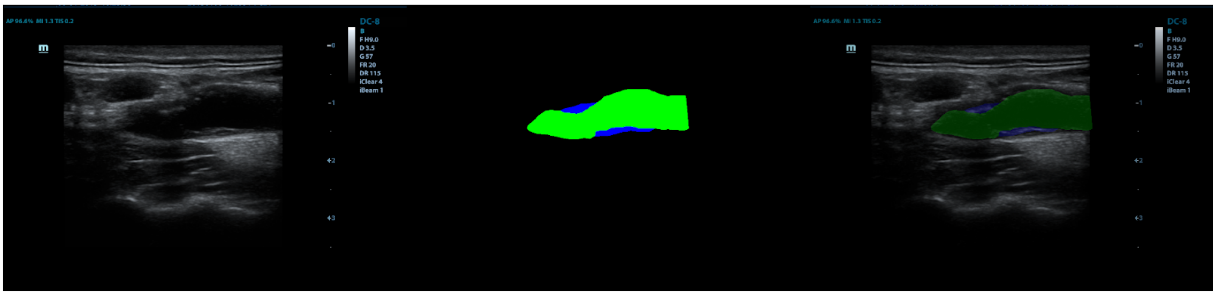


Figure 3.2: At left, we can see the original ultrasound carotid artery image, in the centre, the segmented image and on the right the segmentation overlaid over the original image

It is important to note that we are going to use the equivalence 1 cm = 172 pixels as this is the scale that can be found in our case. We will use K as a constant to define this equality.

## 3.2 Descriptors

To be usable in our machine learning project, ultrasound carotid artery images together with their respective segmentations, must be represented by descriptors. Usually, a descriptor is

* inside space of a tubular structure, such as an artery or intestine.

a number, a vector or a matrix. The choice of a descriptor is important because it has a strong influence on the predictive performance of a model. Descriptors cannot fully describe the ultrasound carotid artery image, but they can draw out relevant features of the image that will help in classification. For the descriptor to be appropriate for the machine learning problem, it has to generate distinct values for the two classes (stable or unstable). This is something that we can not know with certainty at the beginning, but we can help ourselves by drawing on the literature to find out which characteristics would be useful in order to achieve it.

In Table 3.1 we can find a list of the descriptors that will be computed for each type of image.

| **Transversal or cross-sectional** | **Longitudinal** |
|---|---|
| Relative stenosis | Maximum height of lumen |
| Relative lumen area | Median height of lumen |
| Relative lumen diameter | Maximum height of plaque |
| Relative plaque area | Mean height of plaque |
| Haralick texture descriptors wall | Wavelet plaque |
| Wavelet wall | |
| Wavelet plaque | |

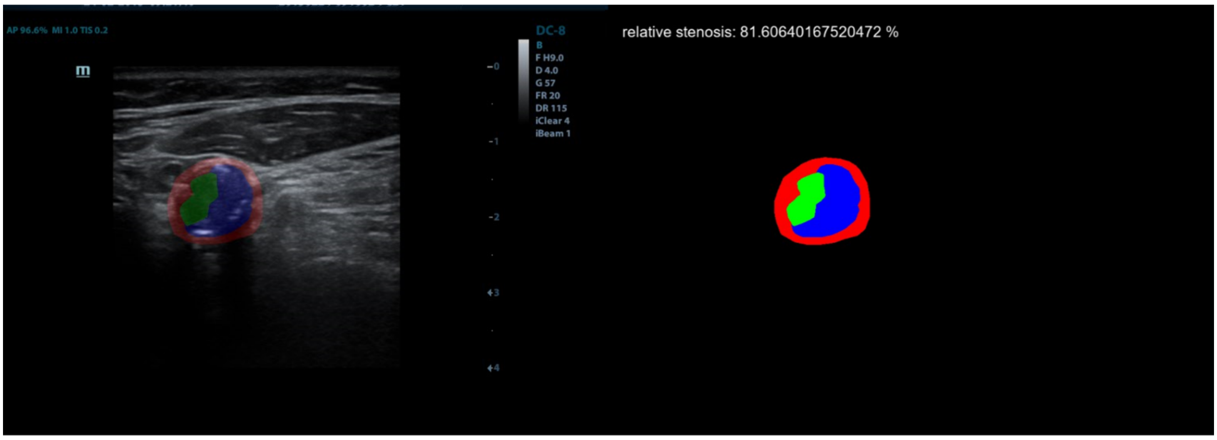
Table 3.1: Descriptors calculated for each type of image

### 3.2.1 Relative stenosis

As we have mentioned in Section 2.2, stenosis is a narrowing of the arteries on either side of the neck that carry blood to the head, face and brain. In this thesis we consider the calculation of the relative stenosis as lumen area vs total vessel area. We can only calculate it in transversal images as the visibility in the longitudinals is different, so we can not do it in the same way. We are defining $T$ as the sum of the pixels of the wall, the lumen and the plaque. $L$ is the total pixels of the lumen. We calculate the percentage of relative stenosis as follows:

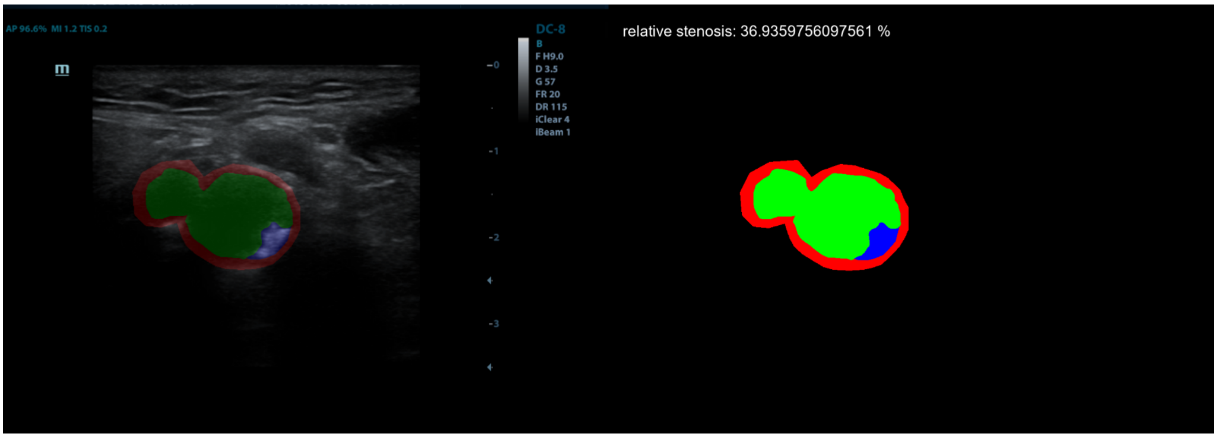
$$S = 100\,(1 - \frac{L}{T}) \tag{3.1}$$

An example is shown in the Figure 3.4 and Figure 3.3. We can see that the results are very different and can mark a difference between the two classes.



relative stenosis: 81.60640167520472 %

Figure 3.3: Calculation of relative stenosis in a cross-sectional unstable plaque



relative stenosis: 36.9359756097561 %

Figure 3.4: Calculation of relative stenosis in a cross-sectional stable plaque

### 3.2.2 Relative lumen area and diameter

These measures will only be calculated with the cross-sectional segmentations. To calculate the relative area of the lumen we use the expression

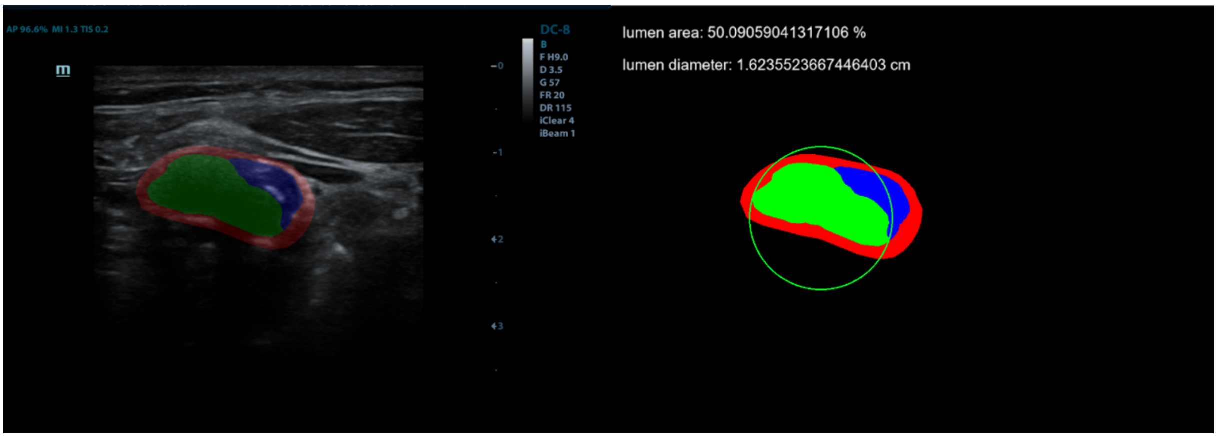
$$A = 100\,(\frac{L}{T}) \tag{3.2}$$

where $L$ is the total pixels of the lumen and $T$ is the sum of lumen, wall and plaque pixels in the segmented image.

To calculate the diameter of the lumen, firstly we are going to obtain the estimated circle around the lumen contour with the function `cv2.minEnclosingCircle()` (see Figure 3.5 and Figure 3.6). This function also gives us the center point and the radius, so we can get the diameter as

$$D = 2\,r\,K \tag{3.3}$$

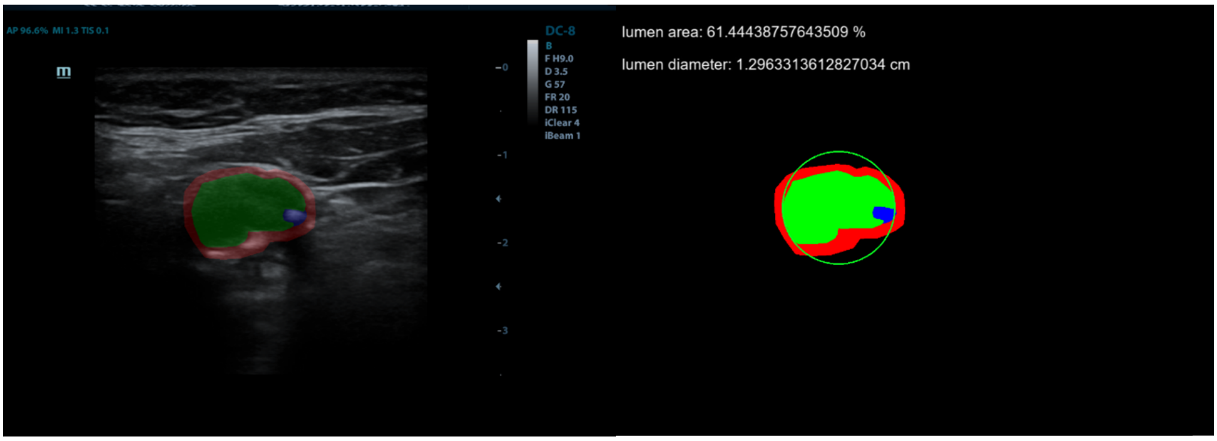
We can see the difference between a stable and unstable plaque in the following images:



lumen area: 50.09059041317106 %

lumen diameter: 1.6235523667446403 cm

Figure 3.5: Relative lumen area and diameter obtained from the segmentation of an unstable plaque



lumen area: 61.44438757643509 %

lumen diameter: 1.2963313612827034 cm

Figure 3.6: Relative lumen area and diameter obtained from the segmentation of a stable plaque

### 3.2.3 Relative plaque area

The percentage of plaque area respect to the whole vessel (wall, plaque and lumen) is calculated in the cross-sectional images. Similar to Section 3.2.2, we use the expression

$$A = 100\,(\frac{P}{T}) \tag{3.4}$$

where $P$ is the total pixels of the colour previously defined as plaque in the segmented image and $T$ is the sum of lumen, wall and plaque pixels of the image.

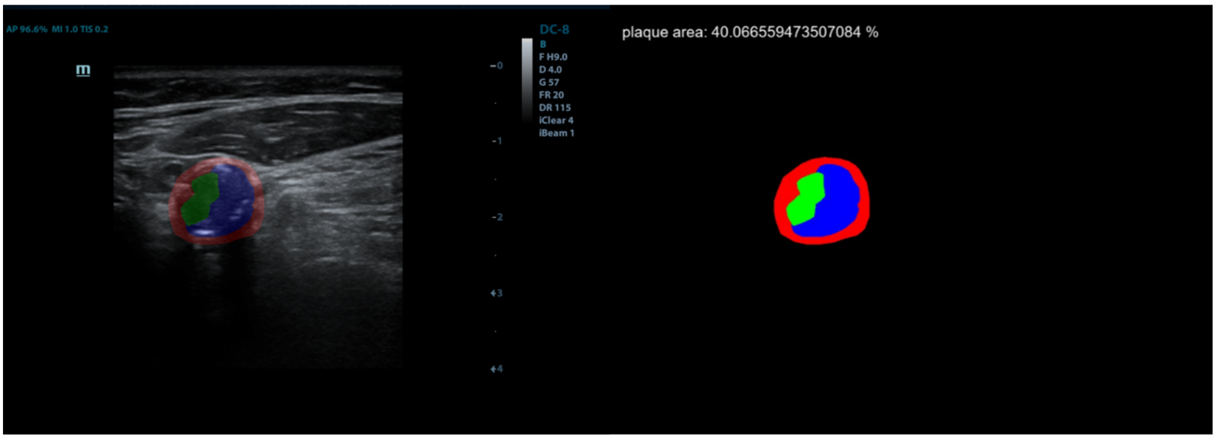


Figure 3.7: Relative plaque area obtained from a cross-sectional unstable plaque

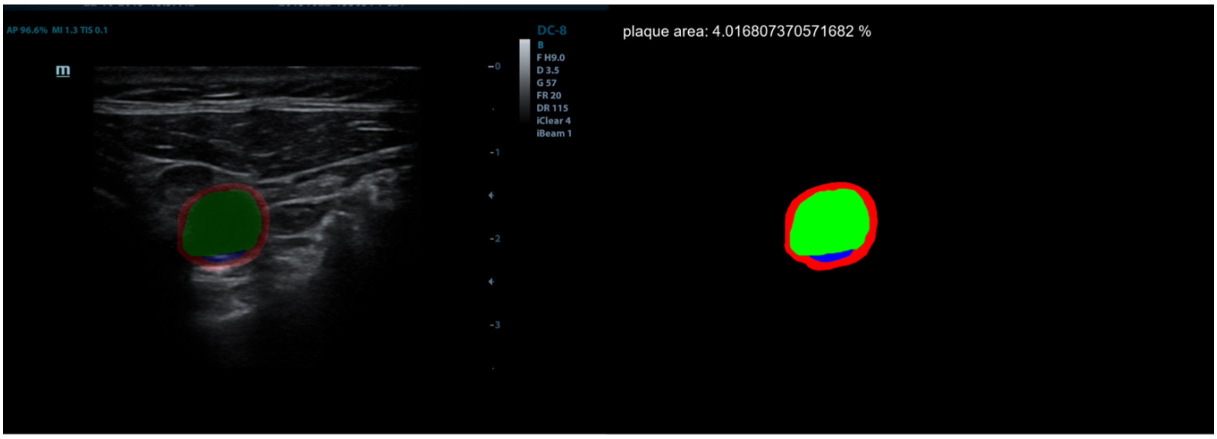


Figure 3.8: Relative plaque area obtained from a cross-sectional stable plaque

The relative area of the plaque is one of the most representative descriptors. It will be really useful in the classification due to its simplicity and effectiveness. It performance can be seen in Figure 3.7 and Figure 3.8 for the two types of classes. In the Figure 3.9 we can find the histogram of both classes, we clearly see that for the unstable plaque the high values predominate compared to the stable ones.

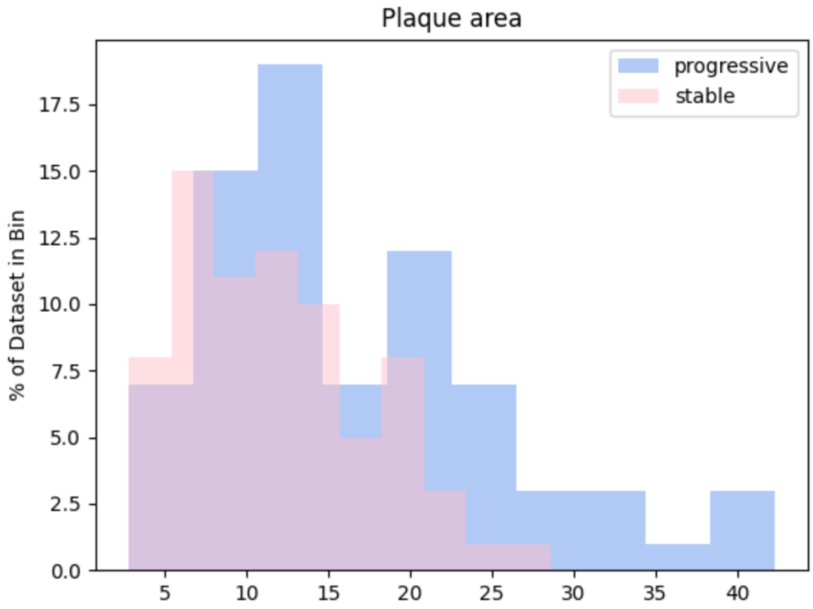


Figure 3.9: Histogram representing relative plaque area (in %) in cross-sectional images

### 3.2.4 Intensity and texture features of the plaque: Haralick

Most texture analysis methods consider the relation between two or more pixels at a time. Haralick, who is Distinguished Professor in Computer Science at Graduate Center of the City University of New York [23], proposed using a gray-level co-occurrence matrix (GLCM) as a method of quantifying the spatial relation of adjacent pixels in an image. This matrix is square with dimension $N_g$, where $N_g$ is the number of gray levels in the image. Element [$i$,$j$] of the matrix is generated by counting the number of times a pixel with value $i$ is adjacent to a pixel with value $j$ and then dividing the entire matrix by the total number of such comparisons made. Each entry is therefore considered to be the probability that a pixel with value $i$ will be found adjacent to a pixel of value $j$ (see Figure 3.10).

$$\mathbf{G} = \begin{bmatrix} p(1,1) & p(1,2) & \cdots & p(1,N_g) \\ p(2,1) & p(2,2) & \cdots & p(2,N_g) \\ \vdots & \vdots & \ddots & \vdots \\ p(N_g,1) & p(N_g,2) & \cdots & p(N_g,N_g) \end{bmatrix}$$

Figure 3.10: Gray-level co-occurrence matrix [8]

Haralick texture features, computed from the GLCM, are widely used due to their simplicity and intuitive interpretations. In our project, we have used the skimage library [14] to calculate the texture properties of the GLCM. Using the function `skimage.feature.texture.greycoprops(P[, prop])`[13] we can obtain six different properties: *contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy* and *correlation* of every image. The equations of this properties can be found in Appendix C.

We will need to calculate the GLCM beforehand. Our aim is to calculate the matrix only with the vessel wall pixels of the original image on the one hand and with the plaque pixels of the original image on the other hand. We are going to apply the mask on the image and set all unwanted pixels to -1, get the desired co-occurring grayscale values ($p$(i,j)) with respect to the offset and filter out all those values containing at least one -1 entry. Co-occurring grayscale values are the set of two pixels defined by the offset which are used for the single comparisons to obtain the co-occurrence matrix. Such, we neglect all values which contain a pixel outside the mask.

We can see an example in the Figure 3.11, where we can find the original image on the left and the mask to be applied to obtain the Haralick descriptors overlaped. At right, we see the result of applying the mask on the original image, where all pixels that are not part of it will be -1 and will not be part of the GLCM calculation.

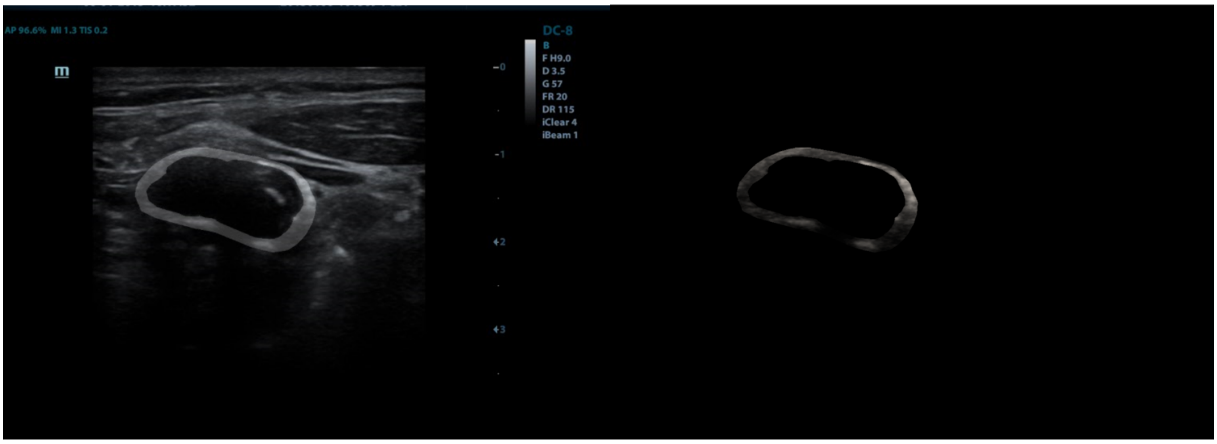


Figure 3.11: At left, an original ultrasound carotid image overlaped with the wall vessel segmentation. At right, the result of applying the binary mask of the wall segmentation on the original image

### 3.2.5 Intensity and texture features of the plaque: Wavelets

The wavelet transform is a well-known signal analysis method in several engineering disciplines. In image processing and pattern recognition, the wavelet transform is used in many applications for image coding as well as feature extraction purposes [2].

In this project, a variation of the discrete wavelet transform is used for characterizing texture properties. This technique is applied to the problems of texture classification and segmentation. The present analysis method, uses an overcomplete wavelet decomposition (the discrete wavelet frame (DWF)) [20] in which the output of the filter banks is not subsampled. Unlike other wavelet-based approaches, this should result in a texture description invariant with respect to translations of the input signal. This property, which appears to be quite desirable in the present context, should yield a better estimation of texture statistics and a more detailed texture characterization at region boundaries.

As in Section 3.2.4, we will apply this descriptor only in the vessel wall and the plaque of the original images, so we need to filter the image to ignore the pixels from the outside of the mask. We are going to follow the same technique of the previous section and apply the segmentation of the desired part as a mask in the original image. Once we have only the pixels that we want, the wavelet filters are going to be calculated.

For simplicity and computational efficiency, we shall use the Haar wavelet with a low-pass filter $H(z) = (1 + z)/2$ and a corresponding high-pass filter $G(z) = (z - 1)/2$.

The main loop is repeated *maxlevel* times. At each level, we filter the input image to provide four sub-bands by using the following filter combinations: $H_x H_y$, $H_x G_y$, $G_x H_y$, $G_x G_y$, where $H_x$ is the low-pass filter applied along the $x$ direction, $G_y$ is the high-pass filter applied along the $y$ direction etc.

For every loop we are going to filter in $y$ direction using the following filters:

- *filterh*: filters all columns of the image by a low-pass Haar filter $H(z) = (1 + z^l)/2$.
- *filterg*: works like *filterh* except the high-pass filter $G(z) = (z^l - 1)/2$ is used.

We are going to use the results to obtain the filtering in the $x$ direction, where the features are the energies in the three high-pass sub-bands for each level. At the last level, the energy of the low-pass band is also added to the output feature vector $v$ [28]. Masking is prior to the application of wavelet filters on the image. We can see the implementation of how to filter in the $x$ direction and obtain the resulting vector $v$ in the following code:

```python
def waveletdescr(self, img, maxlevel=3):
    v = np.zeros(3 * maxlevel + 1)
    for i in range(maxlevel):
        l = 2 ** (i + 1)
        imhy = self.filterh(img, l)
        imgy = self.filterg(img, l)
        v[(3 * i):(3 * i + 3)] = [self.descr(self.filterg(imgy.transpose(),
            l)),
                                  self.descr(self.filterh(imgy.transpose(),
                                      1)),
                                  self.descr(self.filterg(imhy.transpose(),
                                      1))]
        img = self.filterh(imhy.transpose(), 1).transpose()
    v[-1] = self.descr(img)
    return v

def descr(self, x):
    return np.sum(np.power(x, 2)) / x.size
```

This wavelets implementation is an adaptation of the original MATLAB code from the book *Image Processing, Analysis, and Machine Vision: A MATLAB Companion* [28] to the Python language version.

To check that the implementation was correct, we first compared the output of $v$ vector in the two languages (MATLAB and Python) on the same images. Once it was verified that the result was the same, and therefore correct, the mask was applied to the Python version. The pixels that are outside of the mask are going to be ignored in the calculations.

The next step was to check that we could train a classifier with our results and get good accuracy. For this, we used the Brodatz textures. The full implementation can be found at Appendix D. It has been found to work with near-perfect accuracy (97%) for this type of texture.

We have checked that it works and that it should give us good results in our original dataset. This descriptor is one of the most promising ones, as we will see below it gives relevant differences between stable and unstable plaque with one of the highest deviations.

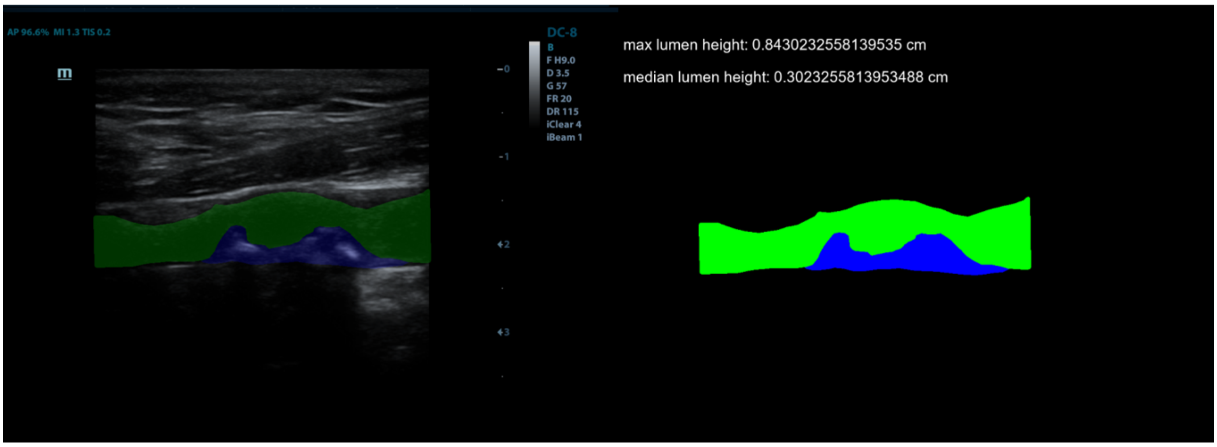
### 3.2.6 Maximum and median height of the lumen

This descriptor is going to be calculated on the longitudinal segmentations. We calculate the median and maximum vertical dimension of the lumen over all columns of the longitudinal images since in the dataset implemented in this thesis the lumen is horizontal.

$$\text{Max} = max\ \{a_i\}_{i=1}^{N}\ K \tag{3.5}$$

$$\text{Median} = \begin{cases} a_{\frac{N+1}{2}}\ K & N \text{ odd} \\ \frac{1}{2}\left(a_{\frac{N}{2}} + a_{\frac{N}{2}+1}\right)\ K & N \text{ even} \end{cases} \tag{3.6}$$

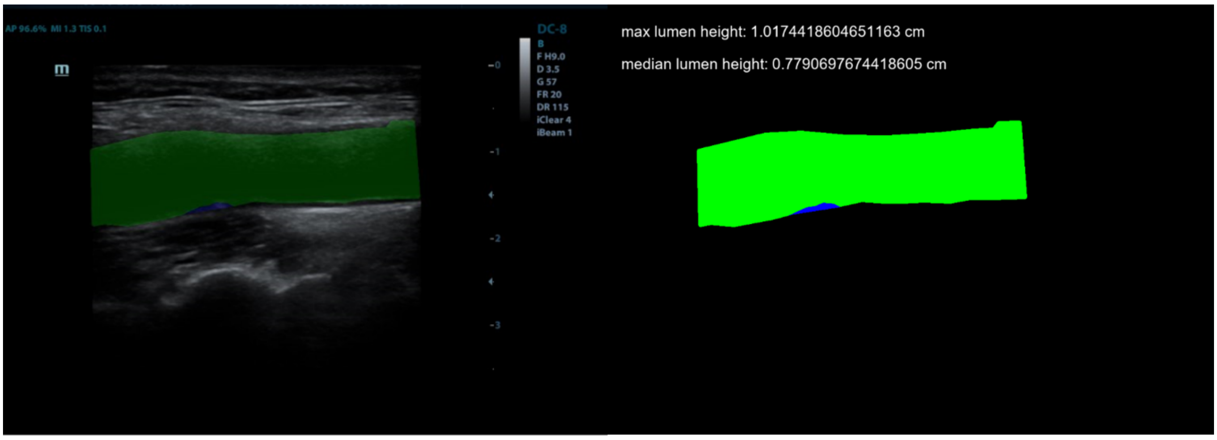
Where $a$ is the number of lumen pixels of the column and the set of values is ordered, $N$ is the total number of columns and $i$ is the index. As before, pixels would be converted to cm using 1 cm = 172 pixels = K. We can see an example of the results in an unstable plaque class image in Figure 3.12 and Figure 3.13 for an stable plaque class image.



max lumen height: 0.8430232558139535 cm

median lumen height: 0.3023255813953488 cm

Figure 3.12: Maximum and median height of lumen obtained from a longitudinal unstable plaque



max lumen height: 1.0174418604651163 cm

median lumen height: 0.7790697674418605 cm

Figure 3.13: Maximum and median height of lumen obtained from a longitudinal stable plaque

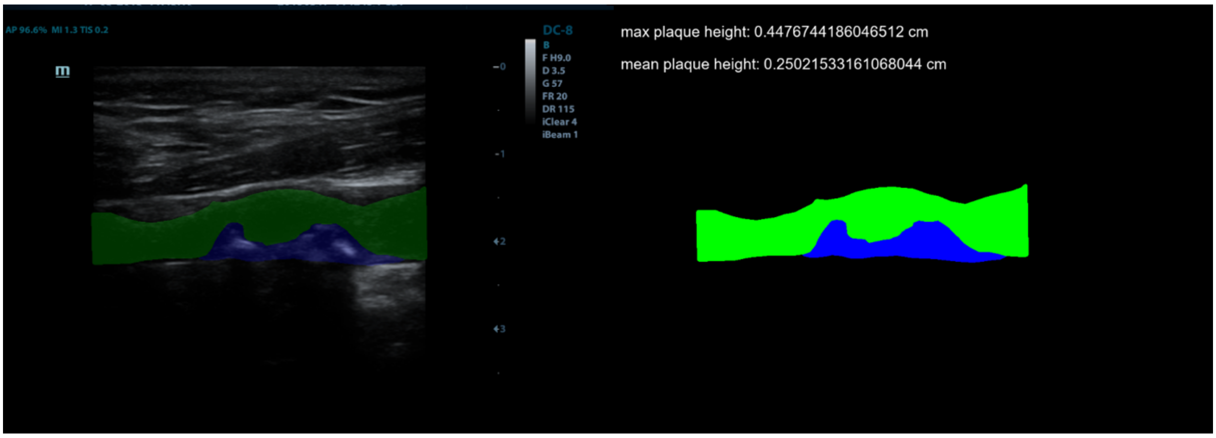
### 3.2.7 Maximum and mean height of the plaque

This descriptor will be very similar to the one in the previous section, except that in this one the calculations will be performed on the plaque and not on the lumen. In this case, the mean will be calculated instead of the median according to the following formulas:

$$\text{Max} = max\ \{b_i\}_{i=1}^{N}\ K \tag{3.7}$$

$$\text{Mean} = \frac{1}{N}\sum_{i=i}^{N} b_i\ K \tag{3.8}$$

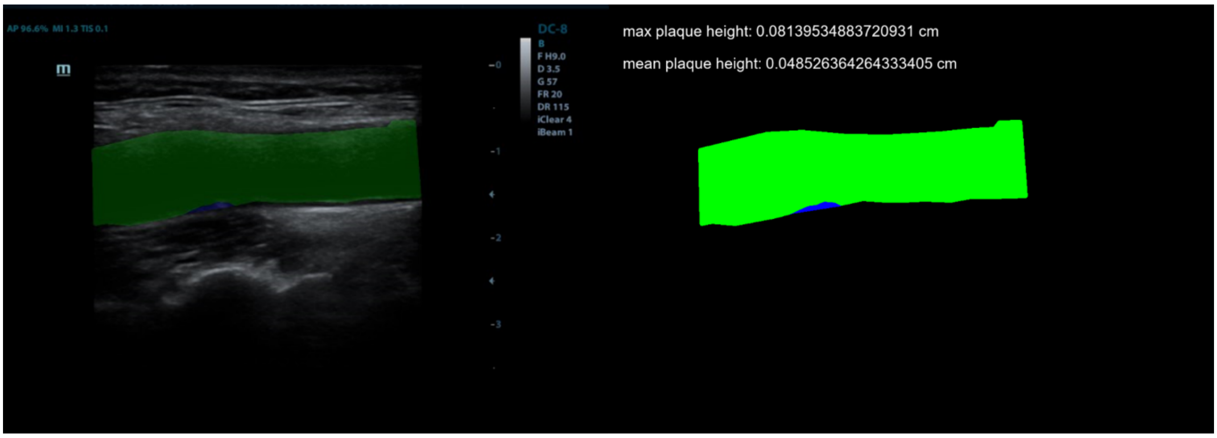
Where $b$ is the number of plaque pixels of the column, $N$ is the total number of columns and $i$ is the index. Pixels would be converted to cm again following the same equivalence. In Figure 3.14 we can find an example of an unstable plaque and in Figure 3.15 an example of a stable one.



max plaque height: 0.4476744186046512 cm

mean plaque height: 0.25021533161068044 cm

Figure 3.14: Maximum and mean height of the plaque obtained from a longitudinal unstable plaque

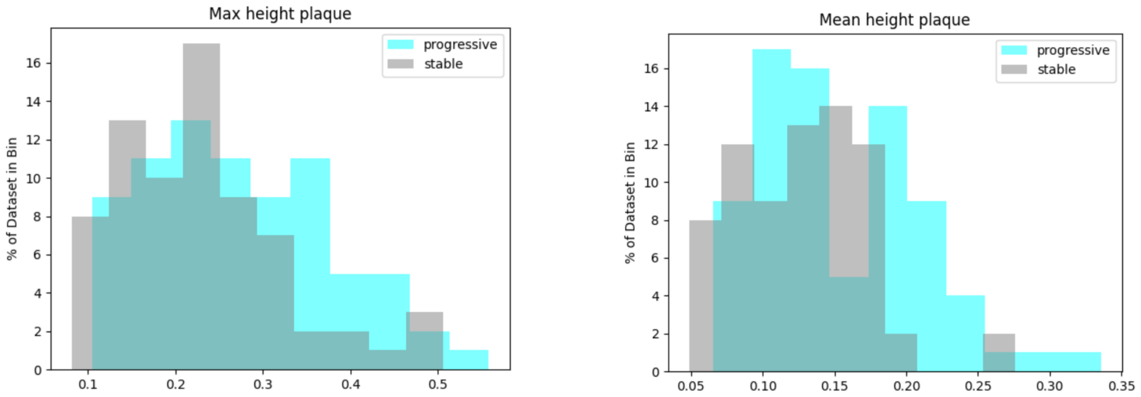


max plaque height: 0.08139534883720931 cm

mean plaque height: 0.048526364264333405 cm

Figure 3.15: Maximum and mean height of the plaque obtained from a longitudinal stable plaque

In Figure 3.16a and Figure 3.16b we can see that again the difference between classes is considerable. For the case of median and maximum, the progressive samples tend to higher values than the stable ones, which is logical since in the unstables the size of the plaque will be bigger.



(a) Histogram representing the max height of the plaque (in cm) in longitudinal images

Histogram representing the mean height of the plaque (in cm) in longitudinal images

## 3.3 Classifying ultrasound carotid artery images

In order to create the machine learning classifier [1], which is the subject of this project, several steps had to be followed to arrive at the final result. In this section we will describe and explain them in detail.

### 3.3.1 Dataset

We start with a dataset which labels are provided in the above-mentioned document (Appendix B). As we can see in Appendix B, in each line we find certain information such as: the name of the image, the patient, the date, whether the image is longitudinal or transversal, the stability and the location of the file. What will interest us most now is to be able to relate the name of the image to its stability so that the image can be labeled. We are going to create an algorithm that will create a dictionary with two keys: 'progressive' and 'stable'. Each of the contents indexed by the keys will be a vector containing the names of the images belonging to the class. Once we have them labelled, the different descriptors that we have seen in Section 3.2 will be calculated, depending on whether they are *transversal* or *longitudinal*. It is important to mention that we will work separately with cross-sectional and longitudinal images throughout the project.

The vectors with the results of the descriptors values of each image will be stored in a .csv file. Finally, we will obtain two .csv files, one for each type of image (longitudinal and transversal). The one for the transversal segmentations will have a total of 42 independent variables for each of the 151 images and the one for the longitudinal segmentations will deal with 15 independent variables for the 149 images.

### 3.3.2 Model implementation

To get the best possible result from our model, we will need a set of metrics to evaluate it and define which the best result is as well as some functions to help us to reach this purpose. We are going to use the method of cross-validation to estimate it. It will run our model several times (as many times as number of divisions of the dataset), and it will calculate the average accuracy of each run. We will be able to get a much more reliable result than if we only do one run each time.

We will also use feature selection, explained in Section 3.3.4, which will allow us to keep only the independent variables that give the best results and get rid of those that are not useful or make the model worse.

Moreover, we will look at the correlations between the different variables, eliminating those with a correlation of more than 0.90 to avoid redundancies (if the variables come from different descriptors). The last step will be to find the classifier, explained in Section 3.3.3, that best fits our model and gives us the highest accuracy. For this purpose, there are tools such as *GridSearchCV* that help us to find the classifier parameters that best fits our model.

### 3.3.3 Classifiers

A classifier [7] is a mathematical function, implemented by a classification algorithm, that maps input vector to one of a small set of classes. Classifiers are a concrete implementation of pattern recognition* in many forms of machine learning.

There are both supervised and unsupervised classifiers [19]. Unsupervised machine learning classifiers are fed only unlabeled datasets, which try to cluster the data, group them such that data within a group is more similar than between groups. Supervised and semi-supervised classifiers are fed training datasets (see Section 3.1), from which they learn to classify data according to predetermined categories.

The aim of this project is to apply the analysis to the ultrasound carotid artery images for a long term development prediction. We will test different classifiers with the most appropriate parameters and evaluate which one gives the best result. In our case, supervised classifiers will be used as the data is labelled.

The following classification algorithms that are going to be described are the ones that will be used throughout the project.

* Pattern recognition is a scientific discipline whose aim is the classification of the objects into a lot of categories or classes. Pattern recognition is also a integral part in most machine intelligence system built for decision making [16].

### Logistic Regression

Logistic regression is a classification algorithm, used when the value of the target variable is binary [29]. Logistic regression is named for the function used at the core of the method, the logistic function. This function, also called the sigmoid function, is a mathematical expression used to map the predicted values to probabilities. It takes any real value and returns values between 0 and 1, labeling them in a binary manner. This function is defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

We can see the representation in Figure 2.2.

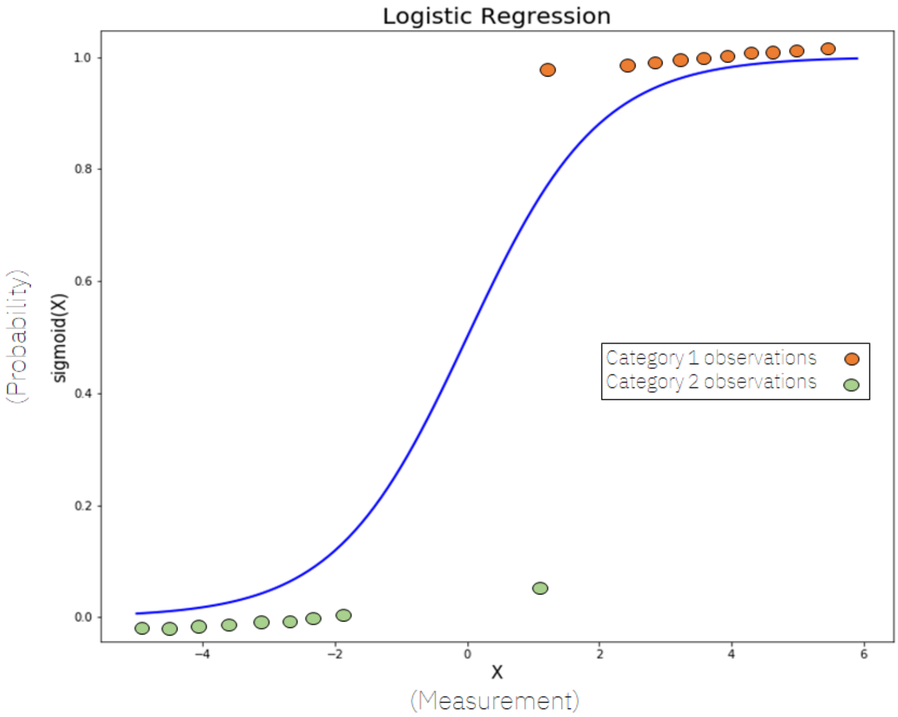


Figure 3.17: Logistic Regression on categorical data from [26]

### Random Forest

Random forests [5] is an algorithm that can be used both for classification and regression. The "forest" it builds, is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of independently trained classifiers improves the overall result. This algorithm adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

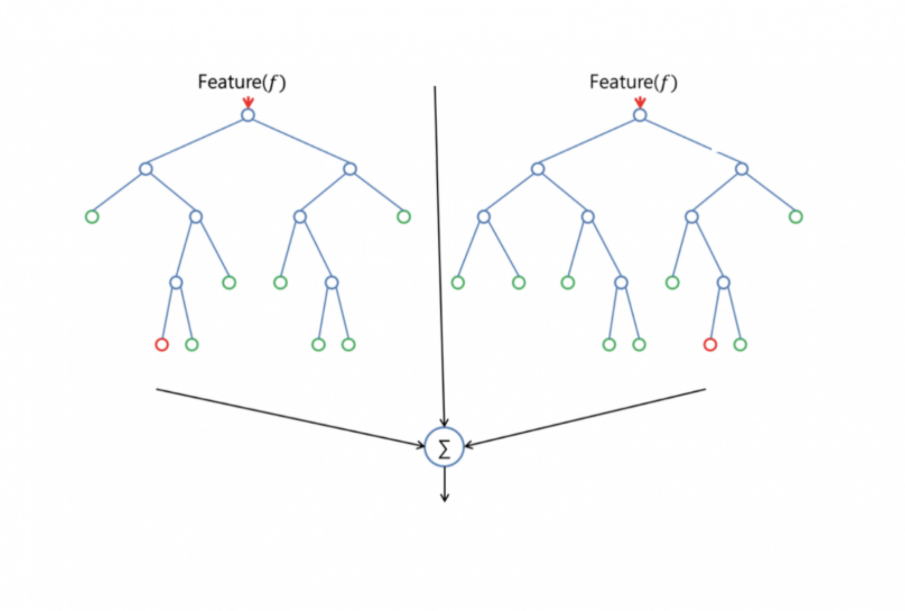


Figure 3.18: Representation of two-tree random forest from [21]

It is considered as a highly accurate and robust method because of the number of decision trees participating in the process and it does not suffer from the overfitting problem, that is why it is a good option for our dataset. We have lots of independent variables that may cause overfitting problems in other classifiers.

### Support Vector Classification

Support Vector Classification (SVC) is an specific type of Support Vector Machine (SVM) that can be used for classification. This classification is achieved by realizing a linear or non-linear separation surface in the input space.

In Support Vector classification, the separating function can be expressed as a linear combination of kernels* associated with the Support Vectors as

$$f(x) = \sum_{X_j \in S} \alpha_j y_j W(x_j, x) + b \tag{3.10}$$

where $x_i$ denotes the training patterns, $y_i \in \{+1, \text{-}1\}$ denotes the corresponding class labels, $S$ denotes the set of Support Vectors, $\alpha_i$ are the corresponding coefficients, $b$ is the offset and $W$ is the kernel function [24].

* Function that takes data as input and transform it into the required form.

### 3.3.4 Improvements in the performance: Feature selection

In this subsection, we are going to try to improve the accuracy of our model by selecting only the features that give us the best performance and eliminating those that can make our model worse. To do so, we are going to resort to some functions that will be useful to us, mainly from the sklearn library. The aim of implementing this selection is to reduce overfitting, improve accuracy and reduce training time. We are going to describe 3 different selection techniques and implement them in our model.

#### Univariate Selection

We will use a function of sklearn library called `sklearn.feature_selection.SelectKBest()` to obtain the 15 features with a strongest relationship with the output variable. In our case we are using the chi-square (chi$^2$) statistical test defined as

$$\tilde{\chi}^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{3.11}$$

for non-negative features to do this selection. Where $O$ is the observed value, $E$ is the expected value and $i$ is the index. In our case $O$ stands for the actual data and $E$ stands for the forecasting result produced by the model. We are obtaining the following results for transversal images:

| | **Specs** | **Score** |
|---|---|---|
| **29** | wavelet_trans_wall_v[9] | 602.799632 |
| **31** | wavelet_trans_plaque_v[1] | 140.636683 |
| **34** | wavelet_trans_plaque_v[4] | 125.950701 |
| **1** | plaque_area | 60.185262 |
| **27** | wavelet_trans_wall_v[7] | 48.870837 |
| **37** | wavelet_trans_plaque_v[7] | 47.867155 |
| **0** | relative_stenosis | 18.613078 |
| **24** | wavelet_trans_wall_v[4] | 12.985258 |
| **26** | wavelet_trans_wall_v[6] | 10.980465 |
| **28** | wavelet_trans_wall_v[8] | 10.502887 |
| **30** | wavelet_trans_plaque_v[0] | 8.043521 |
| **22** | wavelet_trans_wall_v[2] | 7.253864 |
| **25** | wavelet_trans_wall_v[5] | 6.595607 |
| **38** | wavelet_trans_plaque_v[8] | 6.253218 |
| **20** | wavelet_trans_wall_v[0] | 5.225954 |

Table 3.2: Selection of 15 of the best features provided by `SelectKBest` class

As we can see in Table 3.2, wavelets (both plaque and wall), plaque area and relative stenosis are the features that best differentiate the two classes.

### Feature Importance

Feature importance is an inbuilt class that comes with Tree Based Classifiers. We will use `sklearn.ensemble.ExtraTreesClassifier`, which is a class that implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset. Also, it uses averaging to improve the predictive accuracy and control over-fitting.

We are going to get the score for the 10 features of our data that are considered most important or relevant towards our output variable. We can see the result in Figure 3.19.

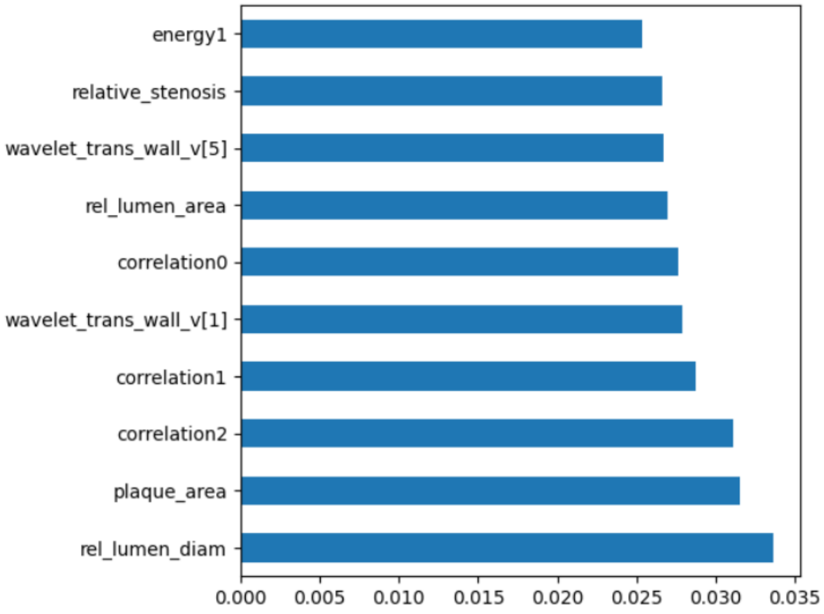


Figure 3.19: Feature importances of tree based classifiers

As expected, wavelets, relative stenosis and plaque area appear. What we did not expect was to find several Haralick texture properties and the relative lumen area and diameter.

### Correlation Matrix with Heatmap

With the correlation matrix we can see how the features are related to each other. We are going to calculate the correlations and print a heatmap as can be seen in Figure 3.20. We can see in the map that the correlation between our features is high in many of them. Shades of green imply high correlations while red shades imply the opposite. Our values are not very decorrelated and therefore we do not obtain very high accuracy results. The aim is to choose only the most useful features in order to make it as decorrelated as possible.

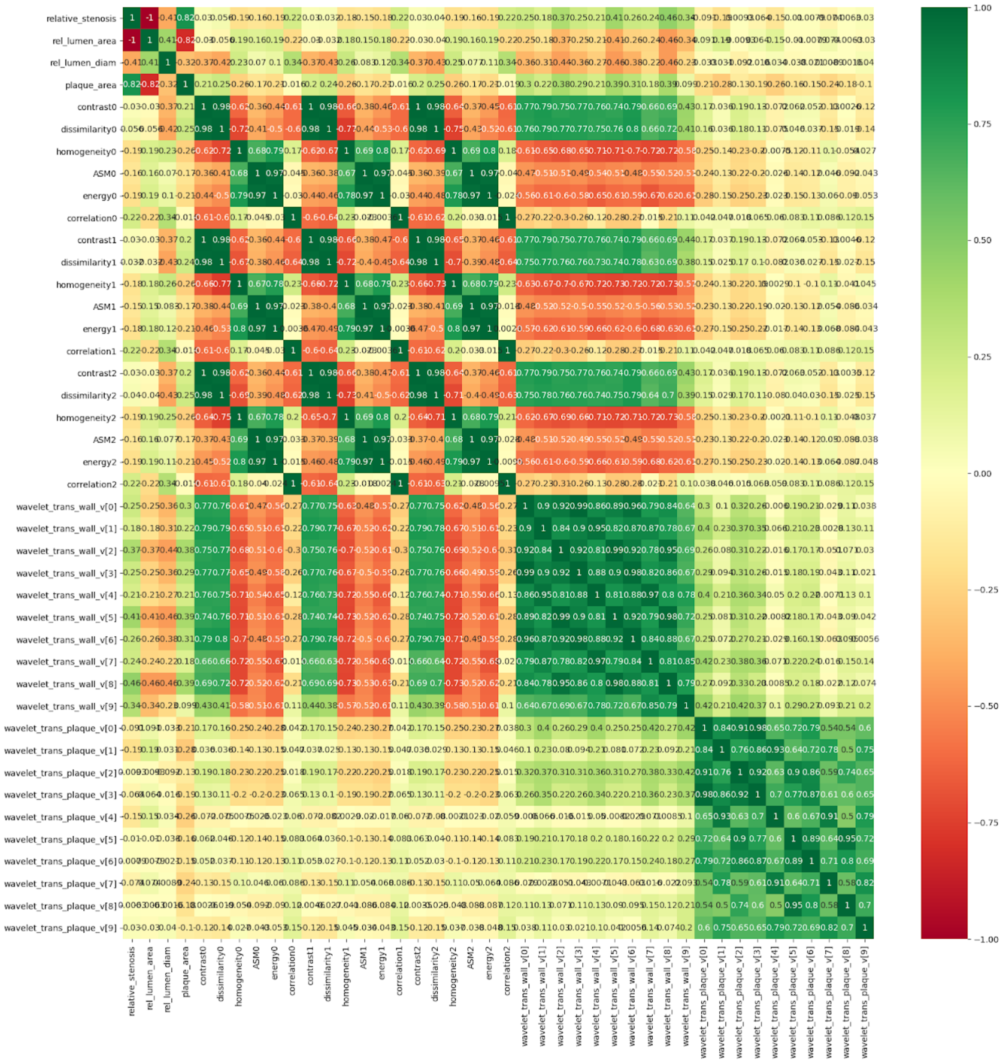


Figure 3.20: Heatmap of the correlations between the features of the transversal dataset

# Chapter 4

# Results

This chapter presents the results obtained from the realisation of the project. Firstly, we are going to explain how the evaluation of the model is done in Section 4.1. Secondly, the evaluation of the desciptors will be carried out, highlighting the most important ones and commenting on their performance in Section 4.2. The next point, in Section 4.3, is going to talk about the performance of ultrasound carotid images classifiers, where we will find the classifier that best suits our model as well as the improvements in the performance thanks to feature selection.

## 4.1 Evaluation process

In order to evaluate our model, we are going to follow a series of steps. We can find the metrics and histograms of all the descriptors that have been implemented in Appendix E. These will give us an idea of which descriptors are the most successful and will offer the best accuracy. Later on, the model will be evaluated on the basis of the accuracy it offers.

Our objective will be to obtain the highest performance, so we will improve our model using feature selection and obtaining the classifier that gives us the highest together with the most suitable parameters for it.

The model is going to be evaluated repeatedly until we find the combination of descriptors, classifier and parameters that gives us the best result.

## 4.2 Evaluation of the descriptors

The evaluation of the descriptors has been done by the computation of histograms and the calculation of basic statistical parameters. The statistical parameters are the following ones:

$$\text{Average mean} = a(x) = \frac{1}{n}\sum_{i=i}^{n} x_i \tag{4.1}$$

$$\text{Median} = m(x) = \begin{cases} x_{\frac{n+1}{2}} & n \text{ odd} \\ \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}\right) & n \text{ even} \end{cases} \tag{4.2}$$

$$\text{Standard deviation} = s(x) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{4.3}$$

Where $x$ is the result of the classifier in the image, $n$ is the total number of images and $i$ is the index.

As mentioned above, all the results are in Appendix E, but we are going to highlight some results in this section.

In Figure 4.1 we can see an example of the performance of the position 9 of the vector $v$ of the plaque of wavelets descriptor. In this case, we can see in the histogram (Figure 4.1) that the difference between the stable and progressive results is big and will create a significant difference for the classification.

Moreover, if we look at Table 4.1, we can see that difference between the average values is considerable as well as the standard deviation.

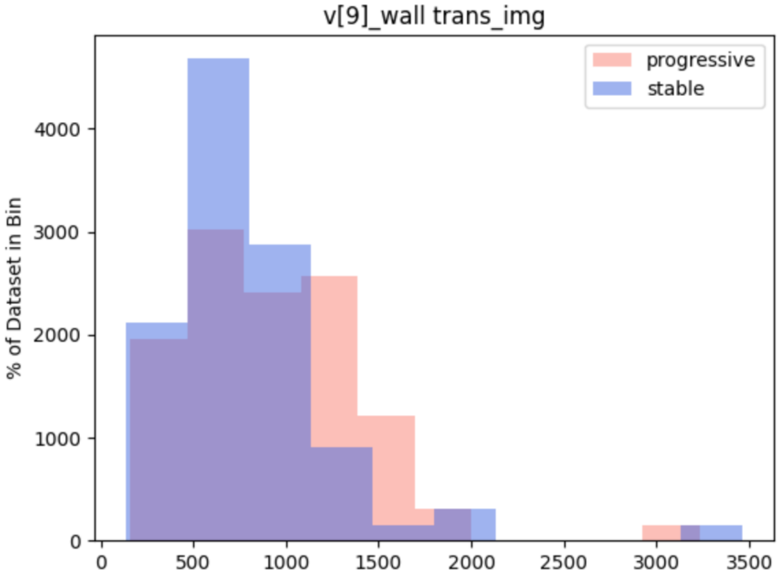


Figure 4.1: Histogram that represents the v[9] of wavelet descriptor in the transversal segmentations of our dataset

| | **Mean** | **Median** | **Standard deviation** |
|---|---|---|---|
| **Progressive** | 918.5524819156045 | 826.5141420807387 | 491.787007900365 |
| **Stable** | 801.2696314708631 | 718.1373682046634 | 474.4778319424617 |

Table 4.1: Statistics of the v[9] of wavelet descriptor in the transversal segmentations of our dataset

We can evaluate the t-test as follows:

$$T\ statistic = t = \frac{(\beta - \gamma)}{\sqrt{\delta^2 + \epsilon^2}} = \frac{(918.55 - 801.26)}{\sqrt{4.56^2 + 4.48^2}} = 18.32 \tag{4.4}$$

Where $\beta$ and $\gamma$ are the mean values of v[9] of wavelet descriptor in the transversal segmentations of progressive and stable images respectively. $\delta$ and $\epsilon$ are the standard errors of progressive and stable images of v[9] of wavelet descriptor as well. The standard error is defined as:

$$Standar\ error = e = \frac{s}{\sqrt{n}} \tag{4.5}$$

With the obtained value in the equation 4.4, we can determine that the difference between the means of the two classes will be significant and that one descriptor will not be enough to distinguish between stable or unstable.

Thanks to the evaluation of the statics (see Appendix E), we will realize that the wavelets descriptor, the relative stenosis and the plaque area are the features that give us a greater difference between the two classes and therefore favour the model to increase the accuracy. This matches what we have seen in Section 3.3.4.

## 4.3 Performance of ultrasound carotid artery images classifier

Initially, all the descriptors explained in Section 3.3.3 were used in the classification and tested with the following results:

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **progressive** | 0.69 | 0.55 | 0.61 | 20 |
| **stable** | 0.59 | 0.72 | 0.65 | 18 |
| **accuracy** | | | 0.63 | 38 |
| **macro avg** | 0.64 | 0.64 | 0.63 | 38 |
| **weighted avg** | 0.64 | 0.63 | 0.63 | 38 |

Table 4.2: Classification report of our model using Linear Regression Classifier in transversal data

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **progressive** | 0.58 | 0.94 | 0.71 | 16 |
| **stable** | 0.92 | 0.50 | 0.65 | 22 |
| **accuracy** | | | 0.68 | 38 |
| **macro avg** | 0.75 | 0.72 | 0.68 | 38 |
| **weighted avg** | 0.77 | 0.68 | 0.68 | 38 |

Table 4.3: Classification report of our model using Random Forest Classifier in transversal data

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **progressive** | 0.82 | 0.38 | 0.51 | 24 |
| **stable** | 0.44 | 0.86 | 0.59 | 14 |
| **accuracy** | | | 0.55 | 38 |
| **macro avg** | 0.63 | 0.62 | 0.55 | 38 |
| **weighted avg** | 0.68 | 0.55 | 0.54 | 38 |

Table 4.4: Classification report of our model using SVC in transversal data

Where `macro avg` is the averaging the unweighted mean per label and `weighted avg` the averaging the support-weighted mean per label.

In this first performance the results were not very clear, in each run the result differed a lot from the previous one and it was difficult to determine which classifier was the best or with what accuracy. Of all of them, it seemed that the Random Forest might be the one that gave us the best result. To solve this issue we will have to move on to the next section.

### 4.3.1 Cross-validation

Now we face the problem that we want to use all the examples for training but also use many examples as an independent test set. To solve that problem we are going to use cross-validation.

With cross-validation we are obtaining a more reliable result. We implemented it in the two most promising classifiers and we obtained the following results for longitudinal images:

| | **Cross-validation** | | | | | **Mean accuracy** |
|---|---|---|---|---|---|---|
| **Random Forest** | 0.5 | 0.4333 | 0.4666 | 0.6666 | 0.6206 | 0.5374 |
| **SVC** | 0.4666 | 0.5 | 0.6 | 0.5 | 0.5862 | 0.5305 |

Table 4.5: Performance of Random Forest and SVC using cross-validation in longitudinal images

It is worth noting that the number of descriptors in the longitudinal images (15) is much smaller than in the cross-sectional images (42). Therefore, in the following subsections we will focus on working with the cross-sectional images, since we have the option to improve the performance. We can not improve much further with the longitudinal images given the current data. As we can see in the Table 4.5, the two classifiers are almost identical and do not give us much information. With the cross-sectional images we are obtaining:

| | **Cross-validation** | | | | | **Mean accuracy** |
|---|---|---|---|---|---|---|
| **Random Forest** | 0.5333 | 0.5 | 0.5333 | 0.4666 | 0.6 | 0.5266 |
| **SVC** | 0.5806 | 0.6666 | 0.7333 | 0.6 | 0.5333 | 0.6222 |

Table 4.6: Performance of Random Forest and SVC using cross-validation in cross-sectional images

What we can conclude from the Table 4.6 is that SVC gives us almost 10% more accuracy than Random Forest and therefore probably we could discard this last one from our options.

### 4.3.2 Grid Search CV

Finally, to completely discard Random Forest as a classifier for our model, we are going to use `sklearn.model_selection.GridSearchCV`. What this function does is an exhaustive search over specified parameter values for an estimator. Given a list of parameters, the function will look for the combination that gives us the best performance. This is the result we are obtaining:

$\{'bootstrap' : False, 'criterion' : 'entropy', 'max\_depth' : 50, 'max\_leaf\_nodes' : 50, 'n\_estimators' : 1000\}$
0.56

As the result obtained is still lower than the result of the SVC in Table 4.6, we can finally discard Random Forest and, from now on, SVC will be the implemented classifier.

### 4.3.3 Final model

By now, the maximum accuracy we have obtained is 0.62 with SVC and with all the available features of the transversal part of the dataset. For the longitudinal images has been 0.53. In the final model we decided to use only the characteristics of wavelets, Haralick texture descriptors, relative stenosis and plaque area as we have seen from Section 3.3.4 that these were the ones that gave us the best results. We are using all wavelets and Haralick texture descriptors because the accuracy was dropping when we were using only the most relevant of the two of them. Now we are obtaining the following accuracy:

| | **Cross-validation** | | | | | **Mean accuracy** |
|---|---|---|---|---|---|---|
| **SVC** | 0.5806 | 0.7 | 0.7 | 0.7666 | 0.6 | 0.6694 |

Table 4.7: Accuracy SVC using cross-validation with transversal datset

In Section 3.3.4 we saw that realtive lumen diameter and area appeared as relevant characteristics. However, when we added them to our model, the accuracy dropped by 60% so they will not be included in the final features.

To sum up, we have managed to improve our classification accuracy by a further 5% with feature selection, obtaining a 67% of accuracy for the cross-sectional images.

# Chapter 5

# Conclusions and future development

The aim of this thesis was to propose and implement a set of descriptors and subsequently a classification model to predict whether the plaque in the carotid artery was unstable (symptomatic) or stable (asymptomatic), using ultrasound carotid artery images and their respective segmentations. A dataset with 149 longitudinal and 151 cross-sectional images and their segmentations has been used.

Coding the descriptors was the most tedious part, as medical images were involved, the implementation had to be detailed and with little margin of error to be considered reliable.

Thanks to the work carried out, it has been possible to determine that the classifier that best adapted to our model was the SVC and we have been able to find, among all the characteristics offered by the descriptors used, which were the ones that gave us the best results.

With feature selection, we were able to go from 62% accuracy to 67% for the transversal images. For the longitudinal images the accuracy was 53%. However, such results are not satisfactory in the field of medical image processing, where these classification model would be used to diagnose the severeness of carotid artery stenosis. Nevertheless, there are many options for improvement which will be explained in Section 5.2.

## 5.1 Limitations

The first limitation that we encounter was the size of the datset. We had to train the model both transversally and longitudinally with approximately 150 images each. This is something that we know will be improved. Martin Kostelanský [18], the author of the segmentations, is working on an automatic segmentation with which we would have approximately 17220 images.

The second limitation was the type of data. Dealing with medical images is complicated because we have to be very accurate in the calculation of the descriptors, as we explained before.

## 5.2 Future work

There are several ways to continue the work of these theses.

Firstly, the implementation of the descriptors could be improved. For example, irregularities in the surface of the plaque or intima-media thickness are two descriptors that have not been implemented and that according to the literature, could be useful in distinguishing classes. Aspect ratio (eigenvalue ratio of the covariance matrix) and circularity (real area over circle area) are possible descriptors to implement as well. More descriptors could also be implemented for longitudinal images, as in this thesis many more cross-sectional than longitudinal descriptors have been considered.

Secondly, the dataset could be improved by obtaining more images and accurate segmentations. It is known that an automatic segmentation will be generated. Using this new segmentation, the dataset will count with approximately 7520 longitudinal and 9700 transversal images. Probably the model would be more reliable. The accuracy of segmentations will also be improved and this will lead to greater precision.

Thirdly, a Convolutional Neural Network (CNN) could be implemented for classification of our data.

# References

[1] Aayushi Johari. "Building your first Machine Learning Classifier in Python". In: (Aug. 2, 2019). URL: https://medium.com/edureka/machine-learning-classifier-c02fbd8400c9.

[2] Adnan Abou Nabout. "Object Shape Recognition Using Wavelet Descriptors". In: (May 16, 2013). URL: https://doi.org/10.1155/2013/435628/.

[3] *Aortic Stenosis Overview.* URL: https://www.heart.org/en/health-topics/heart-valve-problems-and-disease/heart-valve-problems-and-causes/problem-aortic-valve-stenosis (visited on May 9, 2021).

[4] *Atherosclerosis.* May 2, 2019. URL: https://cutt.ly/wnO5keG (visited on May 13, 2021).

[5] Avinash Navlani. *Understanding Random Forests Classifiers in Python.* [Online; accessed May 13, 2021]. 2018. URL: https://www.datacamp.com/community/tutorials/random-forests-classifier-python#algorithm.

[6] *Basic anatomy.* URL: https://cutt.ly/HnO3MHU (visited on May 13, 2021).

[7] James Beckerman. *Classifier.* May 17, 2019. URL: https://deepai.org/machine-learning-glossary-and-terms/classifier (visited on May 13, 2021).

[8] Michael V. Boland. *Haralick texture features.* [Online; accessed June 08, 2021]. Sept. 18, 1999. URL: http://murphylab.web.cmu.edu/publications/boland/boland_node26.html.

[9] C L Dumoulin, H R Hart, Jr. "Magnetic resonance angiography". In: (Dec. 1, 1968). URL: https://doi.org/10.1148/radiology.161.3.3786721.

[10] *Carotid Artery Stenosis.* Feb. 15, 2018. URL: https://cutt.ly/ZnO5WJv (visited on May 9, 2021).

[11] "Cerebral angiography". In: (May 31, 2015). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4638369/.

[12] J Emilio Exaire and Jay S Yadav. "Basic anatomy". In: *Guide to Peripheral and Cerebrovascular Intervention.* Bhatt DL, 2004.

[13] *greycoprops*. [Online; accessed May 20, 2021]. URL: https://scikit-image.org/docs/0.7.0/api/skimage.feature.texture.html#skimage.feature.texture.greycoprops.

[14] *Image processing in Python*. [Online; accessed May 20, 2021]. URL: https://scikit-image.org/.

[15] Iryna Sydorenko. “What You Need to Know About Datasets in Machine Learning”. In: (Apr. 5, 2021). URL: https://cutt.ly/Tn0867s.

[16] Jie Liu,Jigui Sun, Shengsheng Wang. “Pattern Recognition: An overview”. In: (June 2006), p. 61. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1061.1032&rep=rep1&type=pdf.

[17] *Machine Learning*. URL: https://cutt.ly/9n05Kcc (visited on May 9, 2021).

[18] Martin Kostelanský. *Localization and segmentation of in-vivo ultrasound carotid artery images*. Jan. 5, 2021. URL: ftp://cmp.felk.cvut.cz/pub/cmp/articles/kybic/KostelanskyMs2021.pdf.

[19] Tobias Geisler Mesevage. *Classifier$_t$ypes*. Dec. 14, 2020. URL: https://cutt.ly/8n03X5E (visited on May 13, 2021).

[20] Michael Unser. “Texture Classification and Segmentation Using Wavelet Frames”. In: (Nov. 1995).

[21] Niklas Donges. *A complete guide to the random forest algorithm*. [Online; accessed May 13, 2021]. 2020. URL: https://builtin.com/data-science/random-forest-algorithm.

[22] Electrophysiology E. Gregory Thompson MD - Internal Medicine Martin J. Gabica MD - Family Medicine Adam Husney MD - Family Medicine George Philippides MD - Cardiology Rakesh K. Pai MD FACC - Cardiology. *Computed Tomography (CT) Angiogram*. [Online; accessed June 16, 2021]. Aug. 31, 2021. URL: https://www.uofmhealth.org/health-library/bo1097.

[23] *Robert Haralick*. [Online; accessed June 08, 2021]. URL: https://en.wikipedia.org/wiki/Robert_Haralick.

[24] S.V.M. Vishwanathan; M. Narasimha Murty. “SSVM: a simple SVM algorithm”. In: (Aug. 7, 2002). URL: https://ieeexplore.ieee.org/abstract/document/1007516.

[25] Sangeetha Nathaniel, Shreyas Saligram, and Antony Leslie Innasimuthu. “Aortic stenosis: An update”. In: (June 26, 2010).

[26] Sparsh Gupta. *What makes Logistic Regression a Classification Algorithm?* [Online; accessed May 13, 2021]. July 3, 2020. URL: https://towardsdatascience.com/what-makes-logistic-regression-a-classification-algorithm-35018497b63f.

[27] *Stethoscope.* URL: https://en.wikipedia.org/wiki/Stethoscope (visited on May 13, 2021).

[28] Jan Kybic Tomáš Svoboda and Václav Hlaváč. "Chapter 15: Texture". In: *Image Processing, Analysis, and Machine Vision: A MATLAB Companion.* Cengage Learning, 2007, p. 872. ISBN: 0495295957.

[29] *What Is Logistic Regression?* July 9, 2019. URL: https://kambria.io/blog/logistic-regression-for-machine-learning/ (visited on May 8, 2021).

[30] Winnie Fu, Ann Crockett, Gavin Low and Vimal Patel. "Internal Carotid Artery Web: Doppler Ultrasound with CT Angiography correlation". In: (Apr. 2015). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4727644/.

# Appendix A

# Documentation

## A.1 Revision history and approval record

| **Revision** | **Date** | **Purpose** |
|---|---|---|
| 0 | 03/05/2021 | Document creation |
| 1 | 05/05/2021 | Document revision |
| 2 | 10/05/2021 | Document revision |
| 3 | 13/05/2021 | Document revision |
| 4 | 15/05/2021 | Document revision |
| 5 | 20/05/2021 | Document revision |
| 6 | 23/05/2021 | Document revision |
| 7 | 24/05/2021 | Document revision |
| 8 | 31/05/2021 | Document revision |
| 9 | 04/06/2021 | Document revision |
| 10 | 10/06/2021 | Document revision |
| 11 | 15/06/2021 | Document revision |
| 12 | 20/06/2021 | Document revision |

DOCUMENT DISTRIBUTION LIST

| **Name** | **E-mail** |
|---|---|
| Ana Manzano Rodríguez | ana.manzano98@gmail.com |
| Jan Kybic | kybic@fel.cvut.cz |
| Veronica Vilaplana | veronica.vilaplana@upc.edu |

| Written by | | Review and approved by | |
|---|---|---|---|
| Date | 21/01/2021 | Date | 21/01/2021 |
| Name | Ana Manzano Rodríguez | Name | Jan Kybic |
| Position | Thesis Author | Position | Thesis Tutor |

## A.2 Gantt Diagram

The final duration of each task can be seen in Figure A.1, finally more time was spent on the improvements while writing the document, which was started 20 days earlier than was expected in the crtical review.

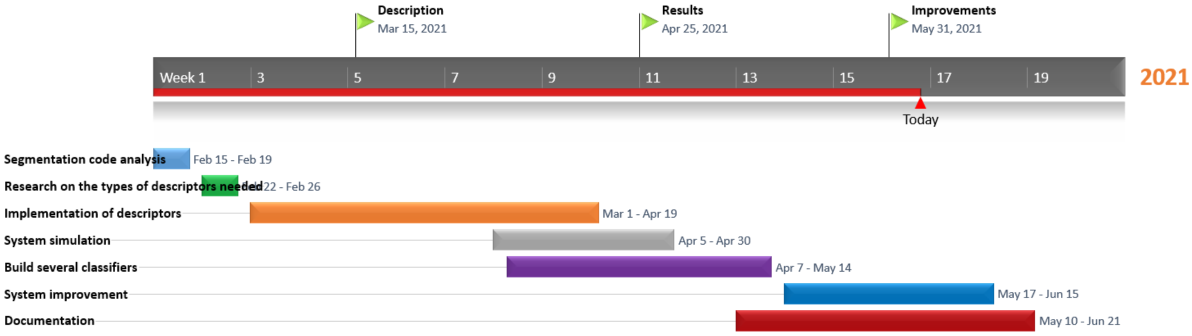


Figure A.1: Gantt diagram

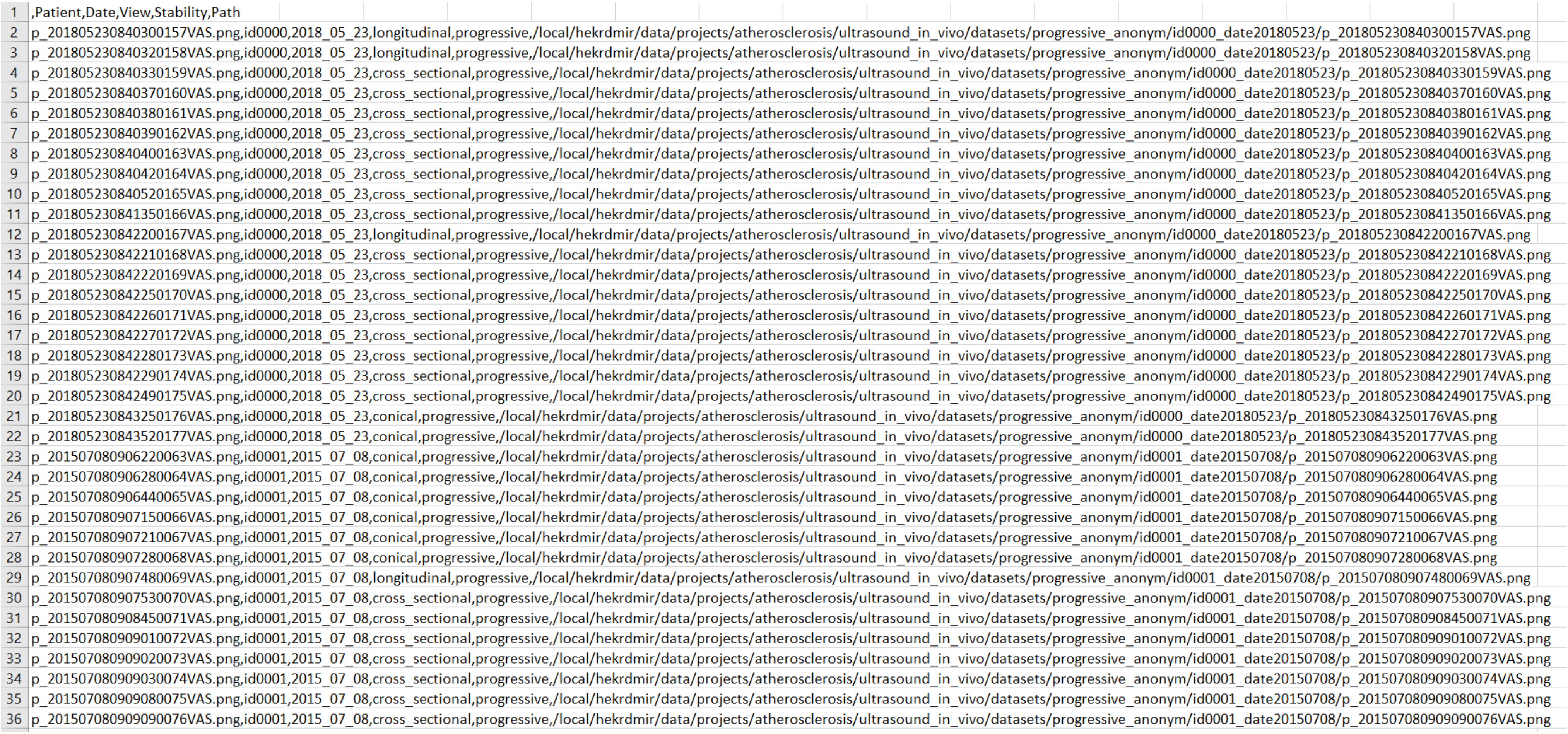
# Appendix B

# Data desciption document

This document called *data_description.csv* has been provided with the segmentations. It is the one that labels our dataset. This document contains the name of each image with its corresponding patient, the date of the sample, if it is longitudinal or transversal, the type of class (stable or progressive) and the path of the image. As explained in subsection 3.3.1, we are going to create an algorithm that relates each image with its class in order to train our model.

In the screenshot of Figure B.1, we can find the first 36 rows of this document, that has a total of 26834 rows.

The document can be found at the following links: Project-Code or `https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/ateroskleroza/-/tree/development/development` or `/datagrid/Medical/ArteryPlaque`.



```
,Patient,Date,View,Stability,Path
p_201805230840300157VAS.png,id0000,2018_05_23,longitudinal,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840300157VAS.png
p_201805230840320158VAS.png,id0000,2018_05_23,longitudinal,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840320158VAS.png
p_201805230840330159VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840330159VAS.png
p_201805230840370160VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840370160VAS.png
p_201805230840380161VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840380161VAS.png
p_201805230840390162VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840390162VAS.png
p_201805230840400163VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840400163VAS.png
p_201805230840420164VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840420164VAS.png
p_201805230840520165VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230840520165VAS.png
p_201805230841350166VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230841350166VAS.png
p_201805230842200167VAS.png,id0000,2018_05_23,longitudinal,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842200167VAS.png
p_201805230842210168VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842210168VAS.png
p_201805230842220169VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842220169VAS.png
p_201805230842250170VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842250170VAS.png
p_201805230842260171VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842260171VAS.png
p_201805230842270172VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842270172VAS.png
p_201805230842280173VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842280173VAS.png
p_201805230842290174VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842290174VAS.png
p_201805230842490175VAS.png,id0000,2018_05_23,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230842490175VAS.png
p_201805230843250176VAS.png,id0000,2018_05_23,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230843250176VAS.png
p_201805230843520177VAS.png,id0000,2018_05_23,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0000_date20180523/p_201805230843520177VAS.png
p_201507080906220063VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080906220063VAS.png
p_201507080906280064VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080906280064VAS.png
p_201507080906440065VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080906440065VAS.png
p_201507080907150066VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080907150066VAS.png
p_201507080907210067VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080907210067VAS.png
p_201507080907280068VAS.png,id0001,2015_07_08,conical,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080907280068VAS.png
p_201507080907480069VAS.png,id0001,2015_07_08,longitudinal,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080907480069VAS.png
p_201507080907530070VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080907530070VAS.png
p_201507080908450071VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080908450071VAS.png
p_201507080909010072VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080909010072VAS.png
p_201507080909020073VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080909020073VAS.png
p_201507080909030074VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080909030074VAS.png
p_201507080909080075VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080909080075VAS.png
p_201507080909090076VAS.png,id0001,2015_07_08,cross_sectional,progressive,/local/hekrdmir/data/projects/atherosclerosis/ultrasound_in_vivo/datasets/progressive_anonym/id0001_date20150708/p_201507080909090076VAS.png
```

Figure B.1: 36 first lines of *data_description.csv* document

# Appendix C

# Equations of Haralick properties

Haralick texture features calculated from GLCMs that have been used in this thesis are listed in Table C.1.

| | |
|---|---|
| **Contrast** | $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$ |
| **Dissimilarity** | $\sum_{i,j=0}^{levels-1} P_{i,j}\lvert i-j\rvert$ |
| **Homogeneity** | $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$ |
| **ASM** | $\sum_{i,j=0}^{levels-1} P_{i,j}^2$ |
| **Energy** | $\sqrt{ASM}$ |
| **Correlation** | $\sum_{i,j=0}^{levels-1} P_{i,j}\left[\frac{(i-\mu_i)\ (j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right]$ |

Table C.1: Equations of Haralick properties [13]

# Appendix D

# Brodatz textures classification using wavelet transform

As has been explained in Section 3.2.5, we have adapted the MATLAB wavelet implementation to Python language. To check that it was working properly, we have used Brodatz textures, which can be seen in Figure D.1. These are the most commonly used texture dataset, especially in the computer vision and signal processing community. In our case we have differentiated 10 classes.

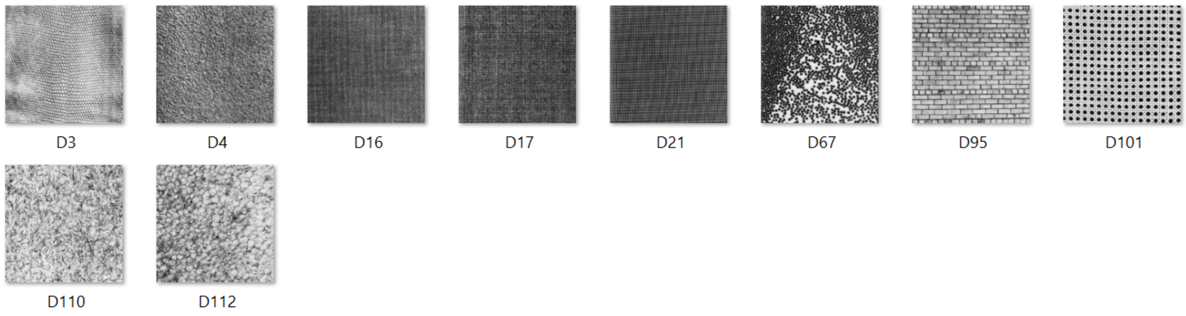


Figure D.1: 10 different types of Brodatz textures

## D.1 Implementation of regular masks

From each image, we have created 25 sub-images of 50x50 pixels each with different coordinates in order to finally have a data set of 260 different images. For every image we have calculated the vector $v$, which contains 10 different coefficients as has been explained before. We have used the Logistic Regression classifier (explained in Section 3.3.3) and we have obtained the results shown in Table D.1.

|  | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **0** | 1.00 | 1.00 | 1.00 | 6 |
| **1** | 0.75 | 1.00 | 0.86 | 3 |
| **2** | 0.88 | 1.00 | 0.93 | 7 |
| **3** | 1.00 | 1.00 | 1.00 | 6 |
| **4** | 1.00 | 1.00 | 1.00 | 9 |
| **5** | 1.00 | 1.00 | 1.00 | 6 |
| **6** | 1.00 | 0.75 | 0.86 | 4 |
| **7** | 0.86 | 0.86 | 0.86 | 7 |
| **8** | 1.00 | 1.00 | 1.00 | 10 |
| **9** | 1.00 | 0.86 | 0.92 | 7 |
| **accuracy** |  |  | 0.95 | 65 |
| **macro avg** | 0.95 | 0.95 | 0.94 | 65 |
| **weighted avg** | 0.96 | 0.95 | 0.95 | 65 |

Table D.1: Classification report of our model using a data set of 260 samples

## D.2 Implementation of irregular masks

With regular images we have seen that the model works, now we create masks with irregular shapes to check that it works correctly with those too. We can see an example in Figure D.2.

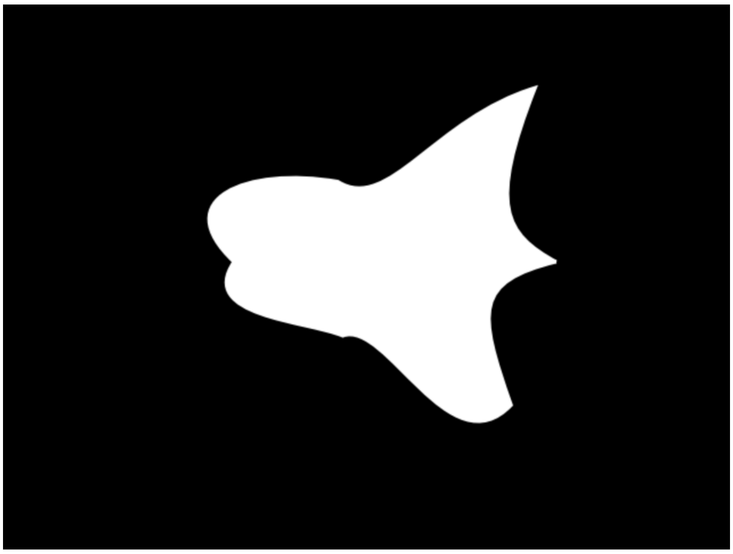


Figure D.2: Example of a random irregular mask

We have created 250 irregular masks that will apply 25 different masks to each of the 10 different textures creating a data set of 250 images. See an example in the Figure D.3, as mentioned before, the part outside the mask will have a value of -1.

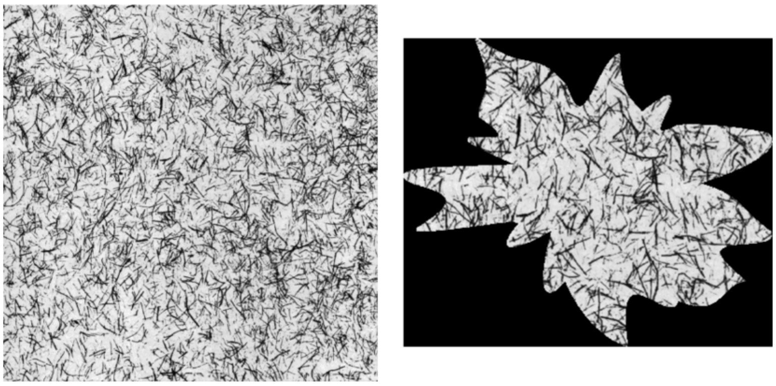


Figure D.3: At left, an example of a texture of Brodatz. At right, the same texture with the irregular mask applied

We used again the Logistic Regression classifier and the classification report with this new data set was obtained. The results can be seen in Table D.2

| | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| **0** | 1.00 | 1.00 | 1.00 | 7 |
| **1** | 0.80 | 1.00 | 0.89 | 8 |
| **2** | 1.00 | 0.86 | 0.92 | 7 |
| **3** | 1.00 | 1.00 | 1.00 | 8 |
| **4** | 1.00 | 1.00 | 1.00 | 4 |
| **5** | 1.00 | 1.00 | 1.00 | 5 |
| **6** | 1.00 | 1.00 | 1.00 | 9 |
| **7** | 1.00 | 0.80 | 0.89 | 5 |
| **8** | 1.00 | 1.00 | 1.00 | 5 |
| **9** | 1.00 | 1.00 | 1.00 | 5 |
| **accuracy** | | | 0.97 | 63 |
| **macro avg** | 0.98 | 0.97 | 0.97 | 63 |
| **weighted avg** | 0.97 | 0.97 | 0.97 | 63 |

Table D.2: Classification report of our model using a data set of 250 samples

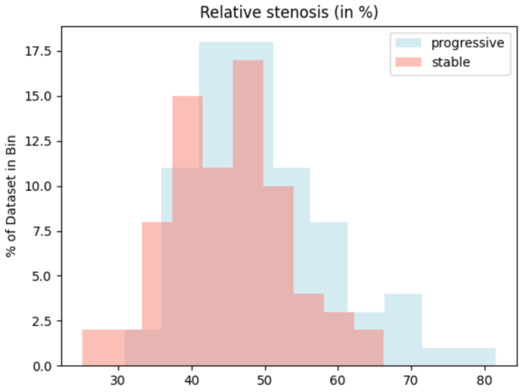
We can now say that the implementation that has been made in this project works well, giving high accuracy results with Brodatz textures.

# Appendix E

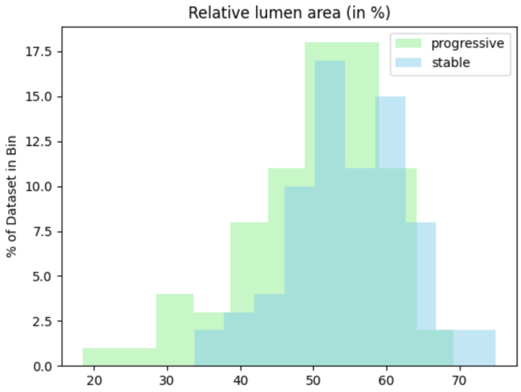
# Histograms and statistics of the descriptors

In this section we will be able to find the histograms and statistics for each of the descriptors. As explained in A, these results will help us to assess whether the descriptor will be useful in our classification or not. Histograms with little difference between the two classes will not be considered useful as well as very similar statistics.

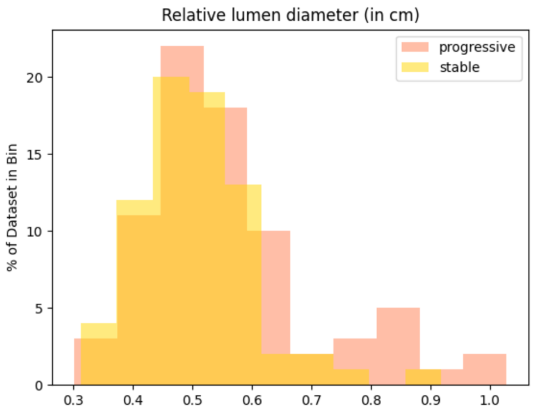
## E.1 Histograms



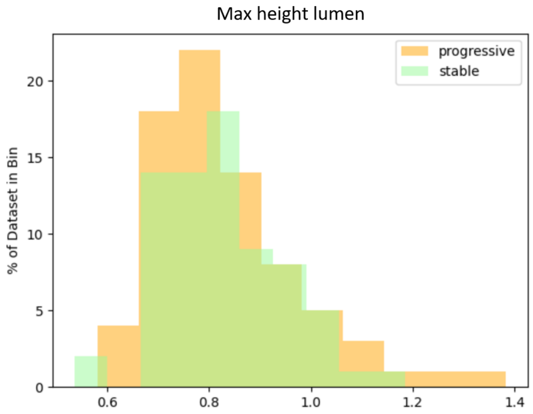
Histogram representing relative stenosis (in %) in cross-sectional images



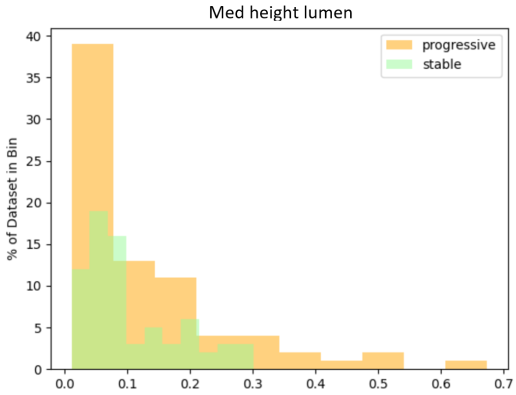
Histogram representing relative lumen area (in %) in cross-sectional images



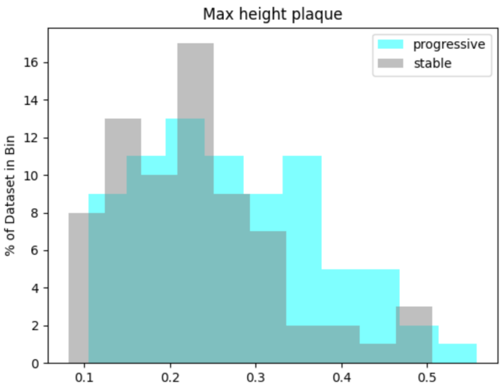
Histogram representing relative lumen diameter (in cm) in cross-sectional images



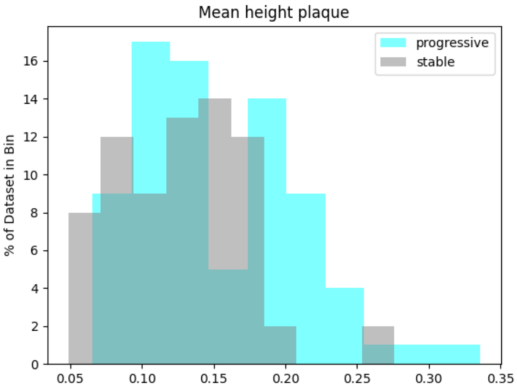
Histogram representing max height of the lumen (in cm) in longitudinal images



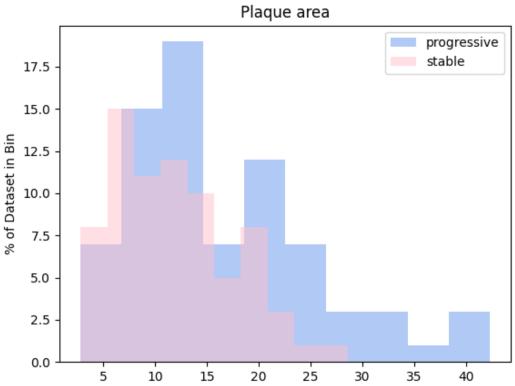
Histogram representing median height of the lumen (in cm) in longitudinal images



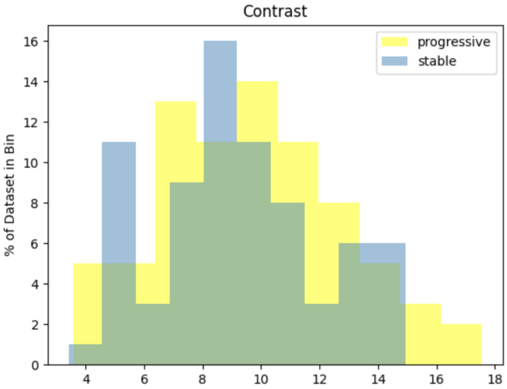
Histogram representing the max height of the plaque (in cm) in longitudinal images



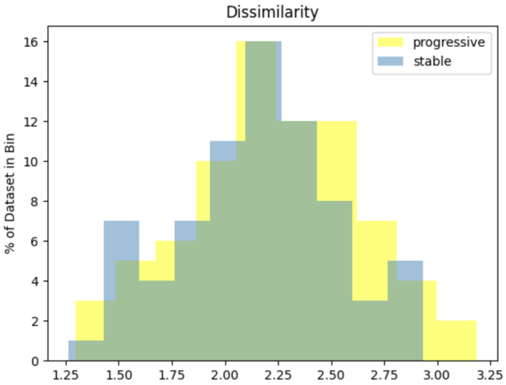
Histogram representing the mean height of the plaque (in cm) in longitudinal images



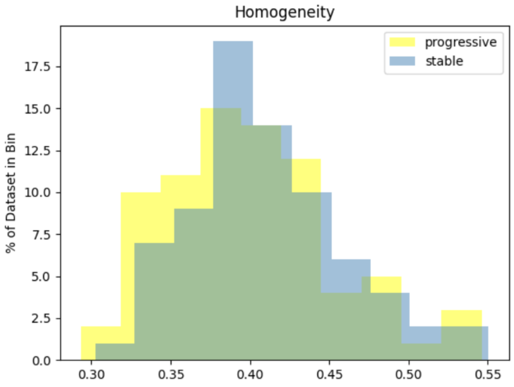
Histogram representing relative plaque area (in %) in cross-sectional images



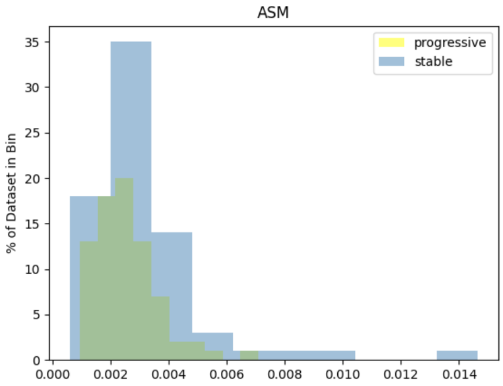
Histogram representing contrast property of Haralick texture descriptors of the wall in cross-sectional images



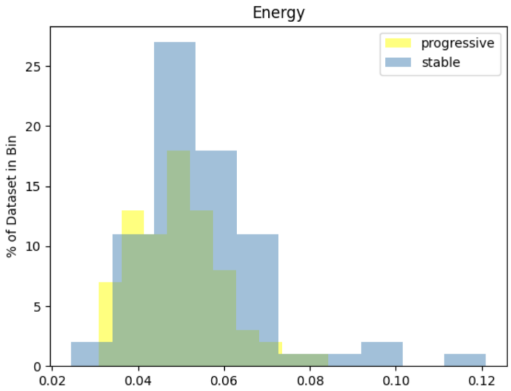
Histogram representing dissimilarity property of Haralick texture descriptors of the wall in cross-sectional images



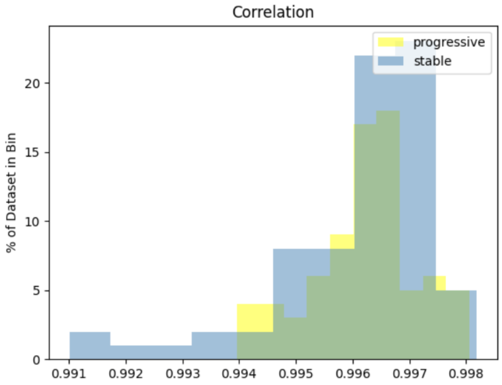
Histogram representing homogeneity property of Haralick texture descriptors of the wall in cross-sectional images



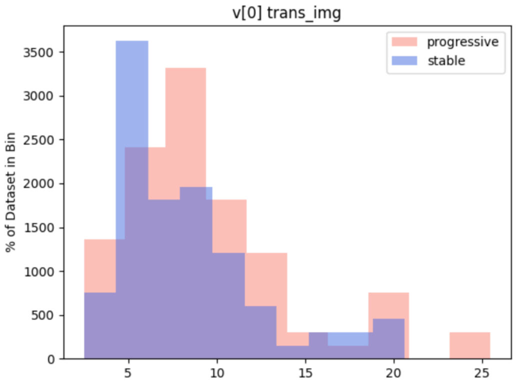
Histogram representing ASM property of Haralick texture descriptors of the wall in cross-sectional images



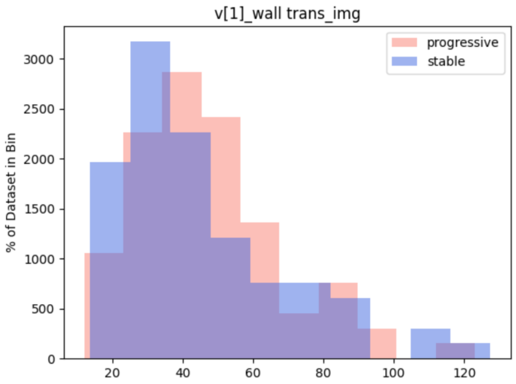
Histogram representing energy property of Haralick texture descriptors of the wall in cross-sectional images



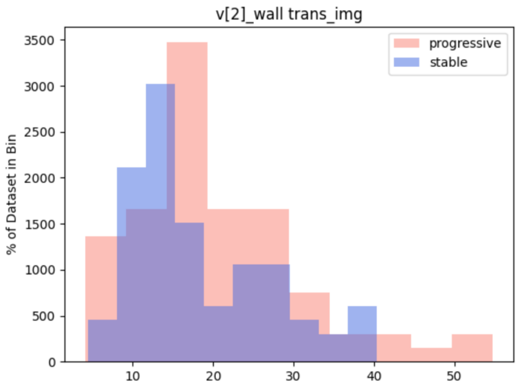
Histogram representing correlation property of Haralick texture descriptors of the wall in cross-sectional images



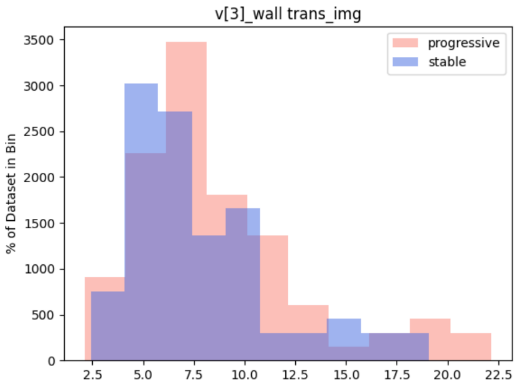
Histogram representing v[0] of wavelet wall vector in cross-sectional images



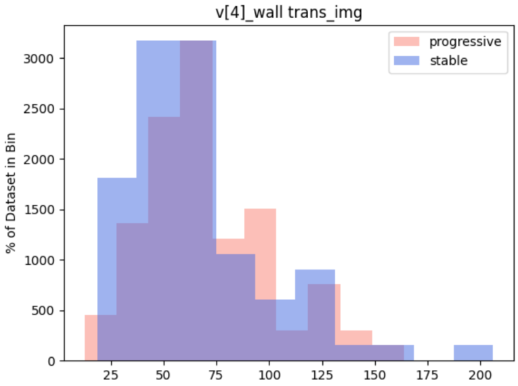
Histogram representing v[1] of wavelet wall vector in cross-sectional images



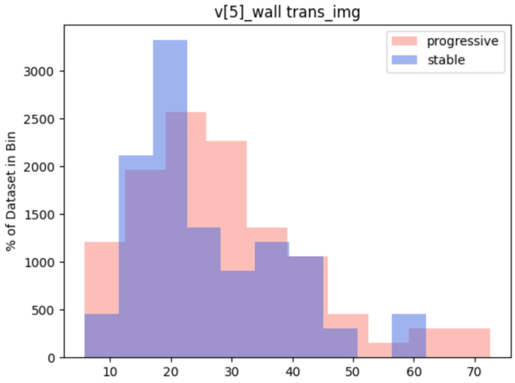
Histogram representing v[2] of wavelet wall vector in cross-sectional images



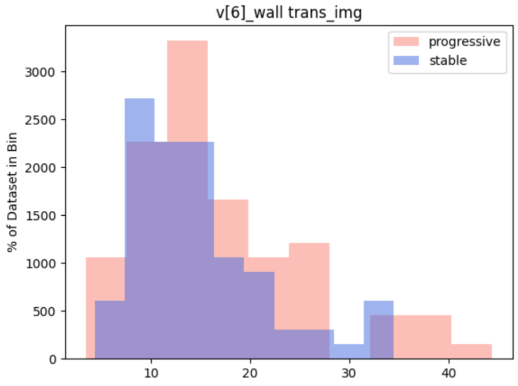
Histogram representing v[3] of wavelet wall vector in cross-sectional images



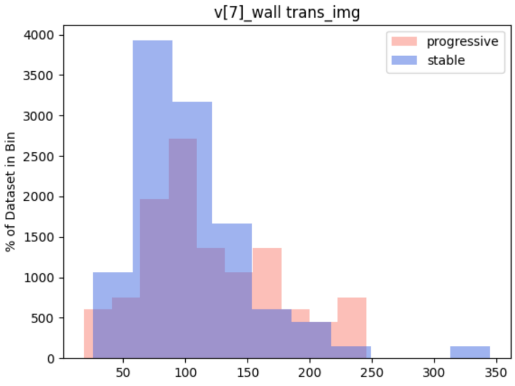
Histogram representing v[4] of wavelet wall vector in cross-sectional images



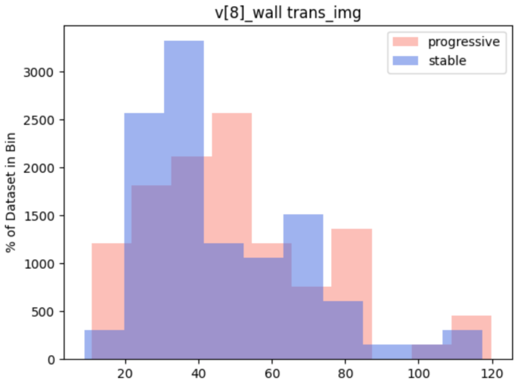
Histogram representing v[5] of wavelet wall vector in cross-sectional images



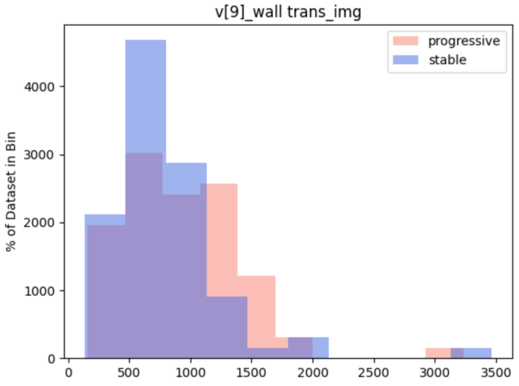
Histogram representing v[6] of wavelet wall vector in cross-sectional images



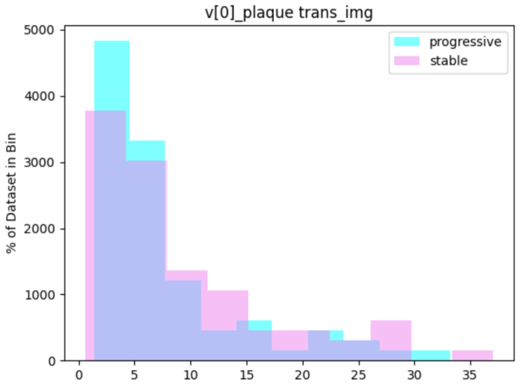
Histogram representing v[7] of wavelet wall vector in cross-sectional images



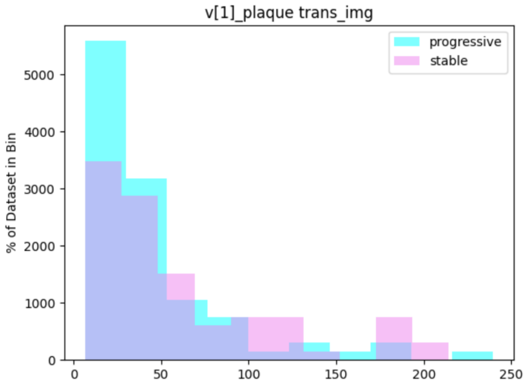
Histogram representing v[8] of wavelet wall vector in cross-sectional images



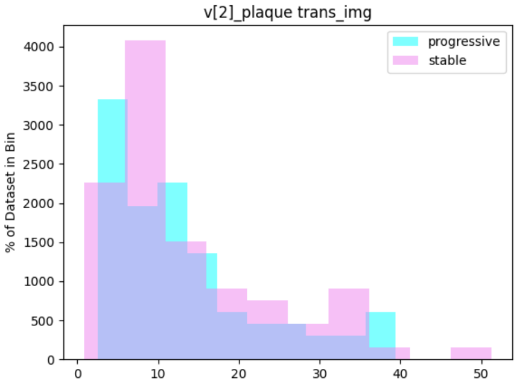
Histogram representing v[9] of wavelet wall vector in cross-sectional images



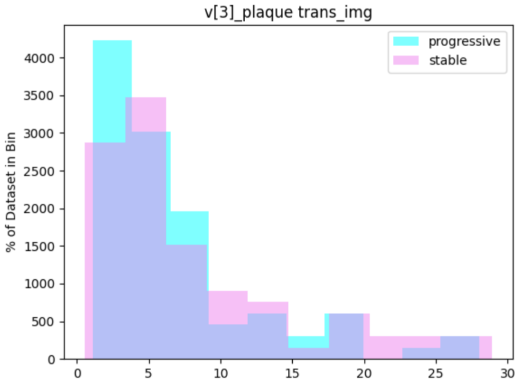
Histogram representing v[0] of wavelet plaque vector in cross-sectional images



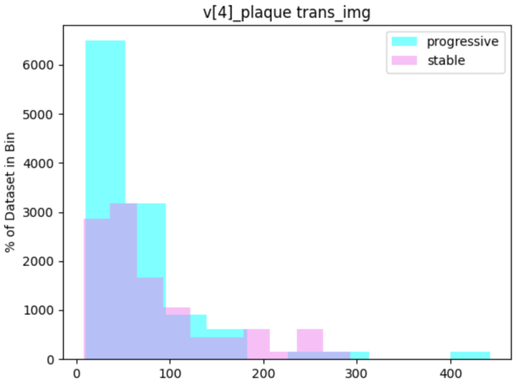
Histogram representing v[1] of wavelet plaque vector in cross-sectional images



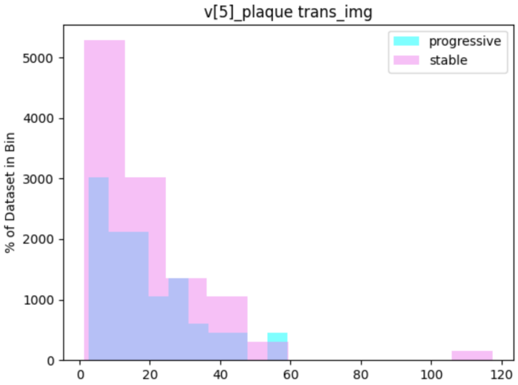
Histogram representing v[2] of wavelet plaque vector in cross-sectional images



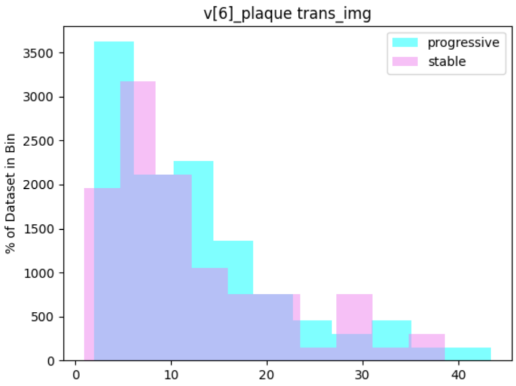
Histogram representing v[3] of wavelet plaque vector in cross-sectional images



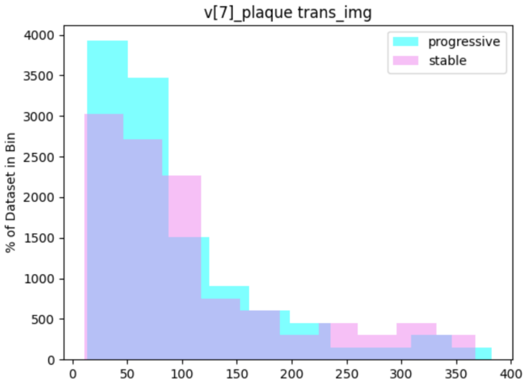
Histogram representing v[4] of wavelet plaque vector in cross-sectional images



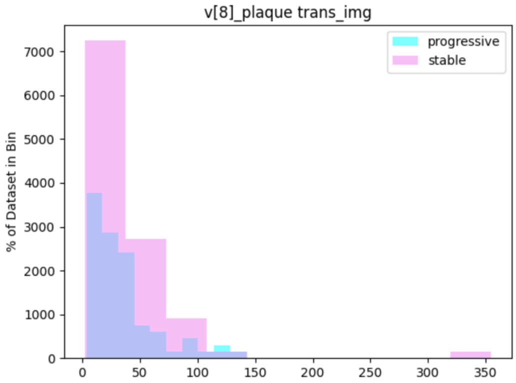
Histogram representing v[5] of wavelet plaque vector in cross-sectional images



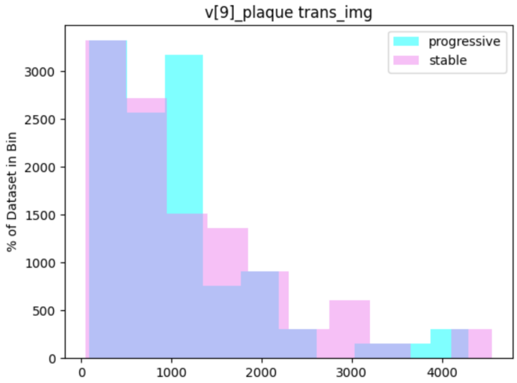
Histogram representing v[6] of wavelet plaque vector in cross-sectional images



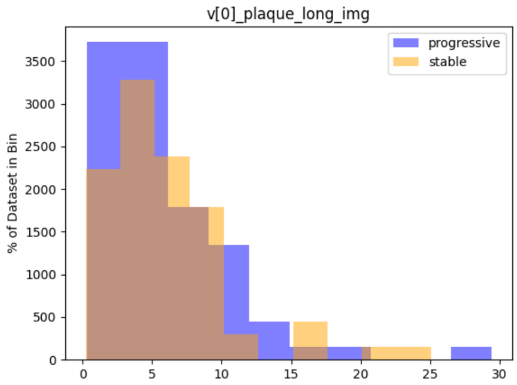
Histogram representing v[7] of wavelet plaque vector in cross-sectional images



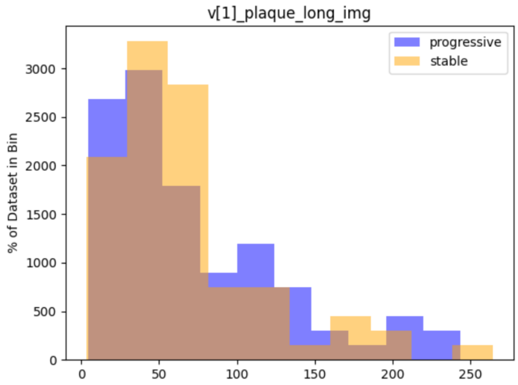
Histogram representing v[8] of wavelet plaque vector in cross-sectional images



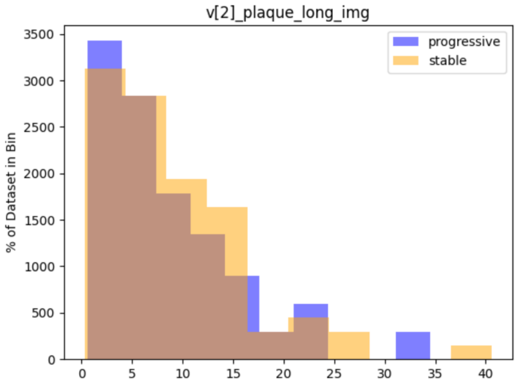
Histogram representing v[9] of wavelet plaque vector in cross-sectional images



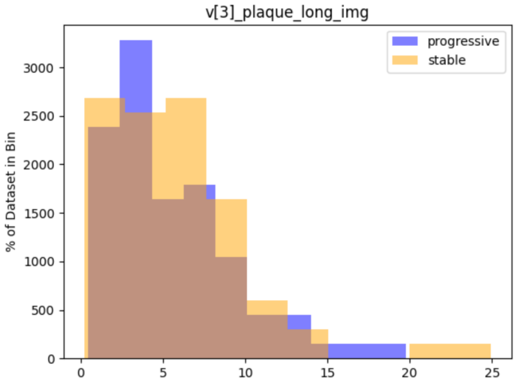
Histogram representing v[0] of wavelet plaque vector in longitudinal images



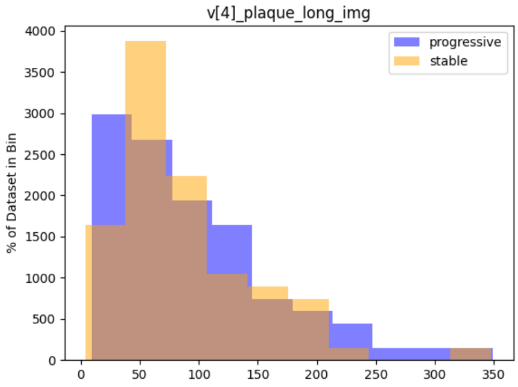
Histogram representing v[1] of wavelet plaque vector in longitudinal images



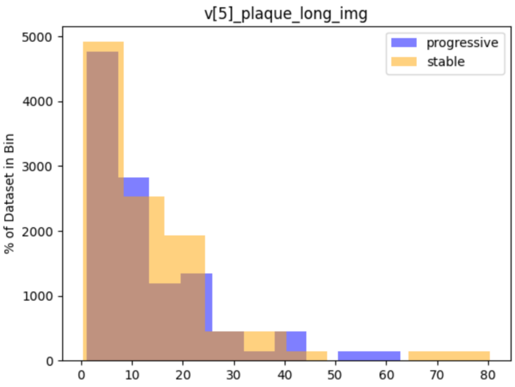
Histogram representing v[2] of wavelet plaque vector in longitudinal images



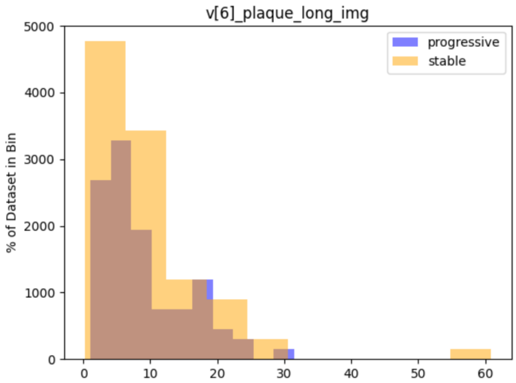
Histogram representing v[3] of wavelet plaque vector in longitudinal images



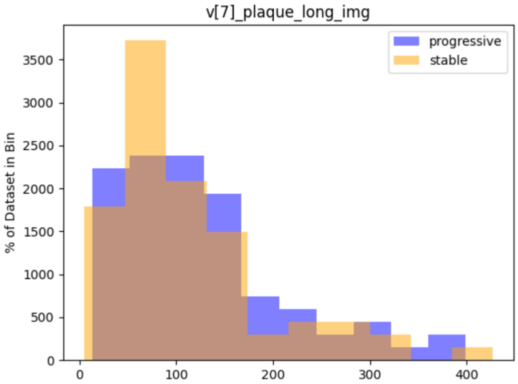
Histogram representing v[4] of wavelet plaque vector in longitudinal images



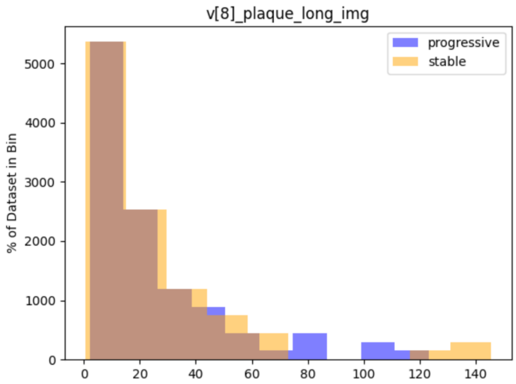
Histogram representing v[5] of wavelet plaque vector in longitudinal images



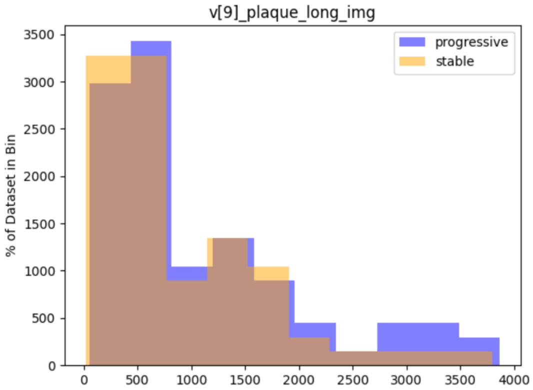
Histogram representing v[6] of wavelet plaque vector in longitudinal images



Histogram representing v[7] of wavelet plaque vector in longitudinal images



Histogram representing v[8] of wavelet plaque vector in longitudinal images



Histogram representing v[9] of wavelet plaque vector in longitudinal images

## E.2 Statistics

| | Mean | | Median | | Standard deviation | |
|---|---|---|---|---|---|---|
| | Progressive | Stable | Progressive | Stable | Progressive | Stable |
| **Relative stenosis** | 49.767% | 44.931% | 48.030% | 44.299% | 9.482 | 8.090 |
| **Relative lumen area** | 50.232% | 55.068% | 51.969% | 55.700% | 9.482 | 8.090 |
| **Relative lumen diameter** | 0.564 cm | 0.504 cm | 0.530 cm | 0.500 cm | 0.147 | 0.099 |
| **Max height lumen** | 0.836 cm | 0.828 cm | 0.802 cm | 0.811 cm | 0.147 | 0.116 |
| **Med height lumen** | 0.128 cm | 0.104 cm | 0.075 cm | 0.075 cm | 0.130 | 0.075 |
| **Max height plaque** | 0.274 cm | 0.229 cm | 0.261 cm | 0.220 cm | 0.105 | 0.093 |
| **Mean height plaque** | 0.153 cm | 0.128 cm | 0.135 cm | 0.129 cm | 0.057 | 0.045 |
| **Plaque area** | 16.634% | 11.859% | 13.629% | 11.100% | 9.253% | 5.816% |
| **Contrast (Haralick text. descr. wall)** | 9.761 | 9.148 | 9.448 | 8.941 | 3.141 | 2.852 |
| **Dissimilarity (Haralick text. descr. wall)** | 2.234 | 2.148 | 2.226 | 2.154 | 0.416 | 0.371 |
| **Homogeneity (Haralick text. descr. wall)** | 0.400 | 0.411 | 0.394 | 0.404 | 0.054 | 0.048 |
| **ASM (Haralick text. descr. wall)** | 0.002 | 0.003 | 0.002 | 0.002 | 0.001 | 0.002 |
| **Energy (Haralick text. descr. wall)** | 0.049 | 0.054 | 0.048 | 0.050 | 0.010 | 0.014 |
| **Correlation (Haralick text. descr. wall)** | 0.996 | 0.966 | 0.996 | 0.996 | 0.000 | 0.001 |
| **Wavelet wall v[0]_trans** | 9.520 | 8.405 | 8.398 | 7.427 | 4.788 | 4.087 |
| **Wavelet wall v[1]_trans** | 46.534 | 45.440 | 42.504 | 38.032 | 20.655 | 23.255 |
| **Wavelet wall v[2]_trans** | 20.355 | 18.424 | 17.490 | 15.236 | 10.397 | 8.676 |
| **Wavelet wall v[3]_trans** | 8.794 | 7.808 | 7.825 | 6.666 | 4.240 | 3.634 |
| **Wavelet wall v[4]_trans** | 71.565 | 66.686 | 68.526 | 59.711 | 31.001 | 33.674 |
| **Wavelet wall v[5]_trans** | 28.253 | 26.074 | 25.819 | 21.945 | 14.104 | 12.230 |
| **Wavelet wall v[6]_trans** | 16.919 | 14.770 | 14.300 | 13.287 | 8.596 | 6.696 |
| **Wavelet wall v[7]_trans** | 117.381 | 105.365 | 104.996 | 100.804 | 52.092 | 51.417 |
| **Wavelet wall v[8]_trans** | 49.742 | 46.089 | 47.456 | 39.116 | 24.394 | 22.020 |
| **Wavelet wall v[9]_trans** | 918.552 | 801.269 | 826.514 | 718.137 | 491.787 | 474.477 |
| **Wavelet plaque v[0]_trans** | 7.742 | 9.080 | 5.271 | 5.736 | 6.801 | 8.138 |
| **Wavelet plaque v[1]_trans** | 47.467 | 61.713 | 31.205 | 42.323 | 43.462 | 52.769 |
| **Wavelet plaque v[2]_trans** | 13.051 | 13.688 | 11.193 | 10.321 | 9.597 | 10.403 |
| **Wavelet plaque v[3]_trans** | 7.200 | 8.089 | 5.134 | 5.596 | 5.846 | 6.690 |
| **Wavelet plaque v[4]_trans** | 67.182 | 82.997 | 48.008 | 59.486 | 65.221 | 68.582 |
| **Wavelet plaque v[5]_trans** | 18.891 | 19.543 | 14.755 | 13.314 | 13.498 | 16.969 |
| **Wavelet plaque v[6]_trans** | 12.364 | 12.178 | 10.296 | 9.159 | 9.029 | 8.975 |
| **Wavelet plaque v[7]_trans** | 95.815 | 107.155 | 71.811 | 81.447 | 76.862 | 87.679 |
| **Wavelet plaque v[8]_trans** | 35.607 | 38.077 | 28.728 | 25.821 | 28.875 | 44.692 |
| **Wavelet plaque v[9]_trans** | 1043.566 | 1131.655 | 897.873 | 739.547 | 857.979 | 998.772 |
| **Wavelet plaque v[0]_long** | 5.875 | 6.050 | 4.449 | 4.981 | 4.738 | 4.663 |
| **Wavelet plaque v[1]_long** | 72.686 | 67.689 | 53.087 | 54.510 | 55.581 | 50.291 |
| **Wavelet plaque v[2]_long** | 8.873 | 9.266 | 6.600 | 7.732 | 6.771 | 7.055 |
| **Wavelet plaque v[3]_long** | 5.535 | 5.834 | 4.517 | 5.183 | 3.815 | 4.358 |
| **Wavelet plaque v[4]_long** | 96.741 | 87.413 | 77.692 | 69.704 | 68.771 | 61.562 |
| **Wavelet plaque v[5]_long** | 13.421 | 14.019 | 8.509 | 10.339 | 12.271 | 14.018 |
| **Wavelet plaque v[6]_long** | 9.106 | 9.823 | 6.776 | 7.084 | 6.326 | 9.066 |
| **Wavelet plaque v[7]_long** | 127.160 | 111.519 | 110.029 | 81.851 | 84.767 | 81.631 |
| **Wavelet plaque v[8]_long** | 25.107 | 24.662 | 15.740 | 14.703 | 26.094 | 28.538 |
| **Wavelet plaque v[9]_long** | 1105.292 | 893.170 | 752.903 | 660.245 | 924.912 | 762.725 |

Table E.1: Statistics table of the descriptors

# Appendix F

# Implementation details

The project was implemented in the programming language `Python`, version `3.8`. The code is documented and can be found in `Github` in the following link: Project-Code or `https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/ateroskleroza/-/tree/development/development`.

The dataset is divided in two folders, `long` and `trans`, where the first one contains the longitudinal images and the second contains the cross-sectional ones. It can be downloaded here: `/datagrid/Medical/ArteryPlaque/in_vivo_platy_segmentace`.

## F.1 Project structure

This project consists of several scripts. Each descriptor has its own script. In *statistics.py* they are all run together with the dataset and *results_output_t5.csv* and *results_output_l3.csv* are created as well as the histograms and statistics. These two files contain the results of all the descriptors for each image, *results_output_t5.csv* for cross-sectional images and *results_output_l3.csv* for the longitudinal ones. These files will be the input to *classifier.py*, which will return the results of how accurate our model is. In *feature_selection.py* we will find the different functions that have been used to make the selection of the best features, the input is going to be again, *results_output_t5.csv* or *results_output_l3.csv* file.