Gonçalves *et al. Journal of the Brazilian Computer Society* (2017) 23:1
DOI 10.1186/s13173-016-0050-7

Journal of the Brazilian Computer Society

RESEARCH Open Access

Applying graphical oracles to evaluate image segmentation results

Vagner M. Gonçalves[1], Marcio E. Delamaro[2*] and Fátima L. S. Nunes[1,3]

Abstract

Segmentation plays an important role in the pattern recognition and image processing areas. Several techniques have been proposed aiming at solving generic issues or particular applications. Traditionally, these techniques have been evaluated by using the Overlap measure, which verifies the coincident and non-coincident areas between the image resulting from a segmentation process and an image considered correct. Albeit widely, this type of measure does not allow flexibility in the assessment process. We here propose an approach to evaluate segmentation techniques using concepts from content-based image retrieval and considering a methodology for testing generic programs with graphical outputs, named graphic oracle. Our approach was applied to evaluate the segmentation of mammographic images, and the results indicate a performance compatible with the traditional measure with more flexibility and precision. Thus, our approach provides a contribution to allow a more flexible segmentation assessment, according to image characteristics and application objectives.

Keywords: Segmentation evaluation, CBIR, Content-based image retrieval, Graphic oracle

Introduction

Segmentation is the process of subdividing an image into its constituent parts or objects in order to isolate a region of interest [1]. Segmentation is essential to most image processing and pattern recognition algorithms as well as applications in related areas.

In approaches involving computer-aided diagnosis (CAD), for example, this task is the basis for locating suspicious regions in various medical imaging modalities. Zheng et al. [2] drew attention to studies showing that the effective segmentation of mammographic masses and microcalcifications are essential for developing CAD schemes. Image segmentation also plays an important role in recognizing biometric measurements [3], objects in satellite images [4], and plant structures in agriculture [5, 6].

A segmentation scheme implemented for a given purpose must undergo an evaluation process to verify its effectiveness in terms of the problem under consideration. This process is largely composed of software testing activities. When a component is executed under specific conditions, the results are observed, and some aspects of the component is evaluated.

Evaluating segmentation algorithms is a special case of testing programs with graphical outputs. This problem has additional complexity as it is no easy task to confirm whether an output image is correct or not according to the system requirements. For example, with respect to mammography CAD schemes, Zheng et al. [2] stated that evaluating automated segmentation schemes is a difficult task and can be ineffective in the case of subtle masses with irregular diffuse edges and surrounded by dense breast tissues.

A systematic review [7] showed that most studies that evaluated segmentation schemes used the Overlap measure in this task [8, 9] and there is no new approach in the CAD context in the literature in the recent years. However, some proposals are cited to evaluate generic segmentation, such as set of scalable discrepancy measures [10], the computation of the difference between

*Correspondence: delamaro@icmc.usp.br
[2]Laboratory of Software Engineering, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Avenida Trabalhador Sancarlense, 400, 13566-590 São Carlos, Brazil
Full list of author information is available at the end of the article

a region extracted from a segmentation map and the corresponding one on an ideal segmentation [11], a metric defined as a function of various error types [12], a measure built according to defined quality criteria, such as shape parameters and homogeneity criterion between regions [13], a metric based on the distance between segmentation partitions [14], and more recently probabilistic metrics [15].

A new software testing technique for programs with graphic outputs was proposed in [16]. This new technique applies concepts of content-based image retrieval (CBIR) to automate the testing oracles. Based on this new approach, this paper aims to propose, implement, and validate a software testing methodology for evaluating image segmentation results. To reach this goal, we use a previously developed framework named O-FIm (Oracle For Images)[1] as a support tool. The validation is conducted with segmentation outputs of a breast region in mammographic images, which is part of a CAD system [17].

The evaluation results from the proposed methodology were compared to the results that used a methodology based on the *Overlap* measure under equivalent test conditions. Thus, the consistency of our methodology could be validated with the results of a traditional measure and thereby determine its advantages and limitations. In general, the results from applying the methodology based on graphic oracles proved to be compatible with those obtained using the methodology based on the Overlap measure. A significant advantage of our approach regards its flexibility, which allows adaptation to the test criteria set for the system under evaluation.

In addition to this introductory section, this paper is organized as follows. The section "Background" offers a background regarding the main concepts used in this article as well as a literature review about assessment of segmentation schemes. The section "Research design and methodology" presents the materials and methods used to carry out the work, which includes a description of the testing methodologies evaluated, the characteristics extracted from the images, the similarity functions used, and the comparison and evaluation of the methods and the results. The section "Results and discussion" presents and discusses the results of the various experiments conducted. Lastly, the section "Conclusions" shows the final remarks.

Background

Segmentation, content-based image retrieval, and graphic oracles are the main concepts used herein, and are presented in the next three subsections. In the section "Evaluation of segmentation," we provide a literature review about segmentation evaluation.

Segmentation

Image processing techniques have been used for several applications in a wide range of knowledge areas. A common classification of such techniques considers three categories: low, intermediate, and high levels. The former category considers techniques to smooth noises and enhance structures of interest in an image. The latter category is responsible for linking the information provided by the previous steps with a knowledge basis. Segmentation is in the second category. It aims at isolating a region of interest in an image. Pixels of this region have common characteristics, which are usually related to aspects of an object represented in the image. Segmentation helps in image classification, since it allows identifying structures present in images.

There is no consensus about the classification of segmentation techniques. One of the mostly used is provided by Gonzalez and Woods [1], which categorizes the techniques in three generic classes: thresholding, edge detection, and region growing. The applicability of each technique depends on the image contrast and on the presence of noise. Segmentation is not a trivial task and must consider the goal of the application as well as the characteristics of the image. This is the reason that explains the constant publication of new specific techniques in the literature.

Content-based image retrieval

Content-based image retrieval is defined by Datta et al. [18] as the technology that assists in the organization of digital images through its visual content. The main components of the systems using this technology are feature extractors, similarity functions, and the image database itself. The feature extractors are used to compose a feature vector for each image. In general, after comparing feature vectors, CBIR systems return the most similar images to a model image provided as a reference, as represented in Fig. 1.

The CBIR system performance and accuracy depend on the choice of suitable features to capture relevant information about the images. This is made in the feature extraction step, which usually occurs after a step of pre-processing and segmentation of the image. From then on, the extraction process assigns values for aspects inherent to the segmented object or the region of interest. For most applications, features for image content representation must be insensitive to variations in size, translation, and rotation. Several extractors can be implemented in a CBIR system, and each of them refers to one aspect of the image. The set of values resulting from the features extraction composes a *feature vector*.

Feature vectors are indexed in the database. Thus, given an user's query image, its feature vector is computed and

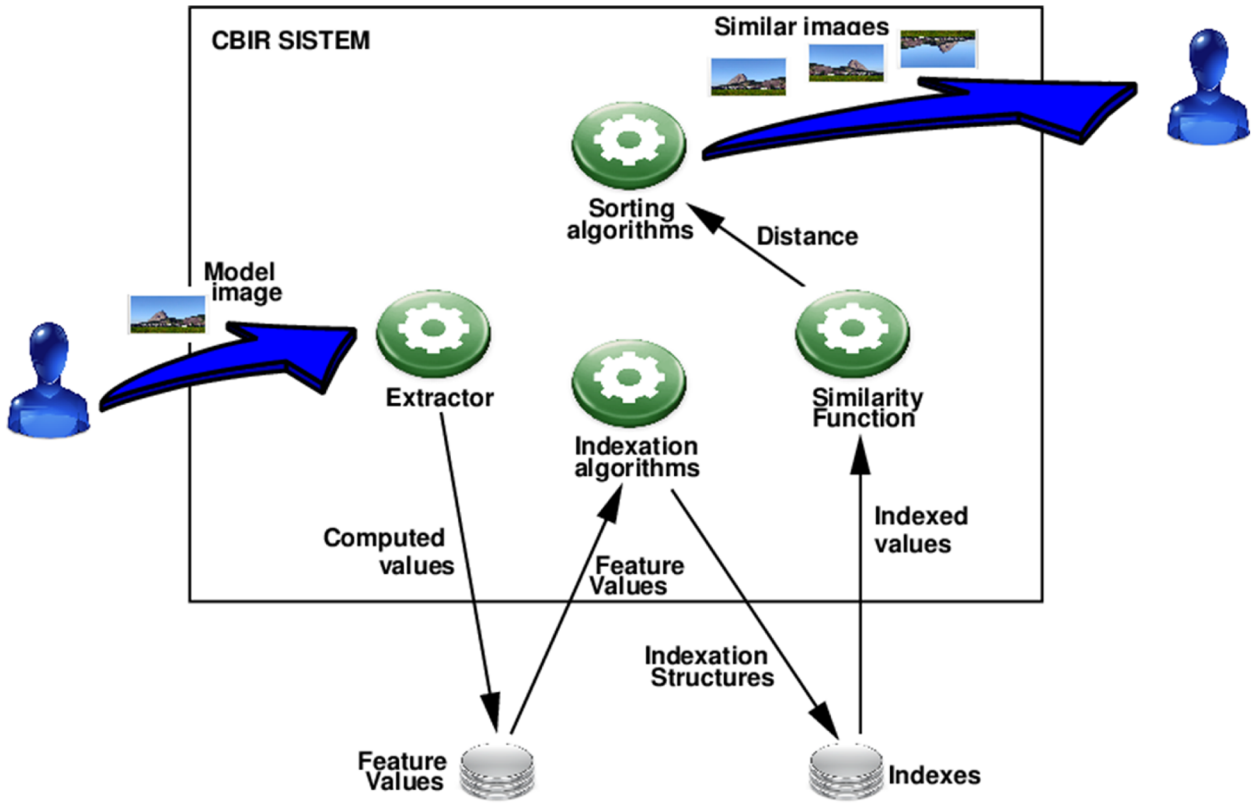

Fig. 1 Scheme of a CBIR system

compared to the feature vectors of images stored in the database [19].

To apply a query per similarity in a image database, it is necessary to measure the distance between the feature vectors by using a similarity function. A similarity function is an algorithm that compares two feature vectors and returns a non-negative value. The most common process for this purpose uses metric distances as, for instance, the Euclidean distance.

The search process can involve the comparison between vectors with high dimensions. Thus, it can be necessary to optimize the performance of the queries by applying adequate index structures, particularly those driven by distance measures. Some works have proposed structures for this purpose [20, 21].

Graphic oracles and the O-FIm framework

The definition of a test oracle refers to an effective mechanism that tells the tester whether the output obtained for a given test is acceptable or not [22, 23]. Oracles are well defined for trivial domains, when inputs and outputs of the program are, e.g., numbers or texts, but for more complex domains, some challenges exist.

A new approach was presented in [16] aiming at contributing to test software with graphic outputs. Using concepts of CBIR, the authors propose to automate the testing oracles for this kind of programs by extracting features from the output images and comparing them with a similarity metric. This approach was named *graphic oracles*.

These oracles can compare one output of the program under test against an image provided as a reference to execute this program. Thus, given the criteria related to the characteristics of the images, defined by the tester, the oracles give a verdict regarding the correct output under examination. Additionally, those researchers proposed a tool to support the definition and use of graphic oracles—the O-FIm framework.

The O-FIm framework is a tool that allows a tester to define features, similarity functions, and additional parameters for graphic oracles. A pre-defined structure was established to help the tester define new descriptors or new similarity functions. Thus, by using a graphic interface, the tester can choose which components will compose the graphic oracle for a program under test. The O-FIm framework meets all the needs to link these

components and returns a verdict to the tester. CBIR concepts were adapted in the aforementioned framework context to allow calculating the distance between the feature vectors of two images, identifying their similarity. The model image refers to an image defined as reference (the expected output), and "image under test" (output of a given program under test) is the one the oracle indicates how similar it is to the model image. Figure 2 shows the framework architecture.

The core of the tool responds to commands to install feature extractors and functions of similarity (plugins) developed for specific image domains. It also provides an application programming interface (API) that allows creating oracles in a simple manner. To conduct a test, the tester must provide a textual description (oracle description) that indicates which are the components installed in the tool (similarity and extractor functions) to be used, as well as their parameters when needed. The O-FIm then uses this description to create the oracle.

The example below shows a textual description example of an oracle compatible with the framework. A graphic oracle is defined by two feature extractors, `MyExtractor` and `OurExtractor`, both receiving the necessary parameters for its implementation. Furthermore, also, as part of the oracle example definition is the inclusion of the Euclidean distance as a similarity function and the definition of a threshold value (`precision`) to indicate the maximum acceptable distance between two compared images to consider them equivalent. In the next sections, we refer to the threshold value in a graphic oracle as *threshold*.

```
----------------------------------------
similarity Euclidian
extractor MyExtractor
      { color = "red" alpha = 78
        rectangle = [100 100 230 240]
      }
extractor OurExtractor
      { rectangle = [0 0 128 64]
        scale = 1.33
      }
precision = 0.46
----------------------------------------
```

The oracle descriptors are a simple way to tell the framework how a graphic oracle should be created in order to carry out a comparison during the execution of a specific test. Given this scenario, the plugins are the tester contributions to create the oracles necessary for the testing activity to be conducted.

Evaluation of segmentation

We conducted a systematic review of CAD systems and metrics to evaluate segmentation in such systems [7]. From a large number of papers retrieved, 10 detailed segmentation techniques and the evaluation metrics used in the testing stage. In this context, evaluation metrics refers to metrics that use quantitative data obtained from a system execution to attribute it a performance index.

Five out of those 10 papers used the Overlap measure to evaluate the segmentation system [8, 9, 24–26]. This measure is the relative intersection area between two regions considered. Considering A_{seg}, the area of an automatically

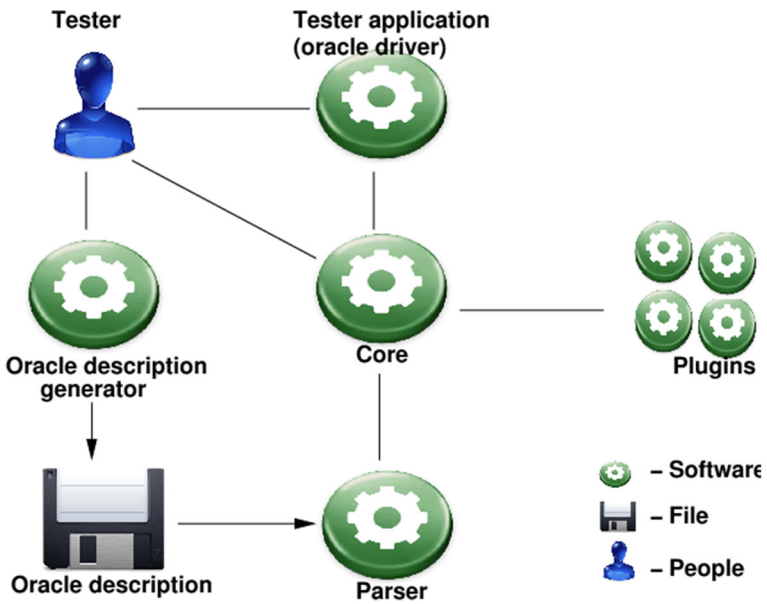

Fig. 2 O-FIm framework architecture (source: [16])

Fig. 3 Example of the original mammogram and respective segmentations. The *first* image shows the original mammogram. The *second* is the image automatically segmented. The *third* and *fourth* images are the results of manual segmentation, performed by two different individuals

segmented region, and A_{man}, the correct area for the segmentation process (manually generated, for example), the Overlap measure is determined by Eq. 1. The value ranges from 0 to 1. The worst performance is indicated by a 0 value, meaning that there is no intersection between the expected area and the area obtained automatically. Otherwise, value 1 indicates a perfect segmentation.

$$\text{Overlap}\left(A_{\text{seg}}, A_{\text{man}}\right) = \frac{|A_{\text{seg}} \cap A_{\text{man}}|}{|A_{\text{seg}} \cup A_{\text{man}}|} \tag{1}$$

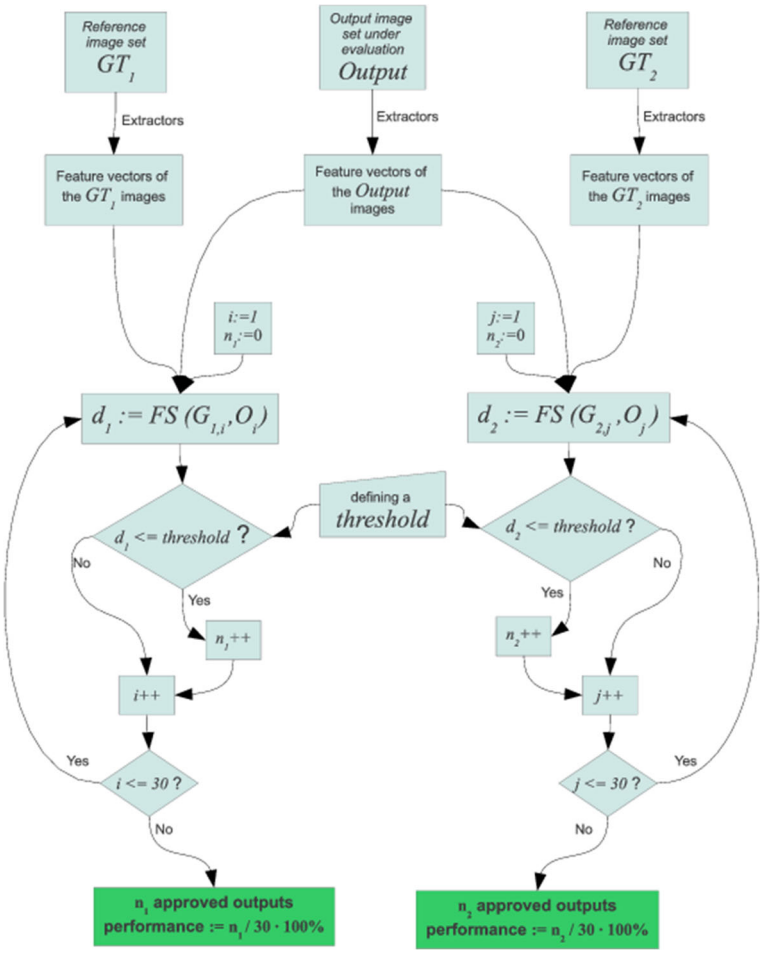

Fig. 4 Evaluation methodology based on the graphic oracles applied to evaluate the breast segmentation scheme

The *relative area difference* measure, implemented in the evaluation of the works [2, 26], measures the extent of the automatically segmented region that does not coincide with the expected correct region. This measure is defined in Eq. 2. If $A_{\text{seg}} = A_{\text{man}}$, then the relative area difference is equal to zero.

$$\text{Relative area difference}\left(A_{\text{seg}}, A_{\text{man}}\right) = \frac{|A_{\text{seg}} - A_{\text{man}}|}{|A_{\text{man}}|} \tag{2}$$

Other metrics also used in the works included were *specificity* and *sensitivity* [27–29]. These metrics are part of a set of very traditional statistic metrics used for evaluating CAD systems. They are based on the concepts of true positive diagnoses (TP), true negatives (TN), false positives (FP), and false negatives (FN) [30]. When applied to segmentation, an approach to use these metrics is to determine the pixels for TP (belonging to the region of interest and segmented), TN (do not belong to the region of interest, and were not segmented), FP (do not belong to the region of interest but were segmented), and FN (belong to the region of interest and were not segmented).

The metrics presented in this section refer to the generic approaches that can and are used to evaluate segmentation schemes developed for different types of applications. These metrics were introduced at this time so that they could be compared using the approach proposed herein.

Research design and methodology

We proposed, applied, and evaluated a methodology based on graphic oracles to test a segmentation scheme used in [17]. We used the O-FIm framework as base technology to conduct all the experiments. The validation of the proposed methodology was performed by comparing its results with the results of a methodology based on the Overlap measure. Thus, we verified to what extent the graphic oracle approach based on CBIR can contribute to an effective evaluation of segmentation algorithms, as well as its advantages and disadvantages when compared to methods based on metrics as those presented in the section "Evaluation of segmentation".

First, we defined the case study and a graphic oracle was built for evaluating images of this case study. Next, we chose the similarity functions and the suitable features to be extracted from the images. The next step was to implement these artifacts in the O-FIm framework and to define a method to evaluate segmentation. Lastly, we compared the results with an Overlap-based evaluation approach.

Case study

The segmentation algorithm used in our tests automatically isolates the breast region in mammograms. The objective of this case study was to compare the results of an automated segmentation process with the results of a manual segmentation considered correct. Gray-level mammographic images were processed considering the steps described as follows. Firstly, the image is analyzed to find the center of mass and discover on which side the breast image is located (right or left). If the breast is located on the left side, a rotation is executed to put the

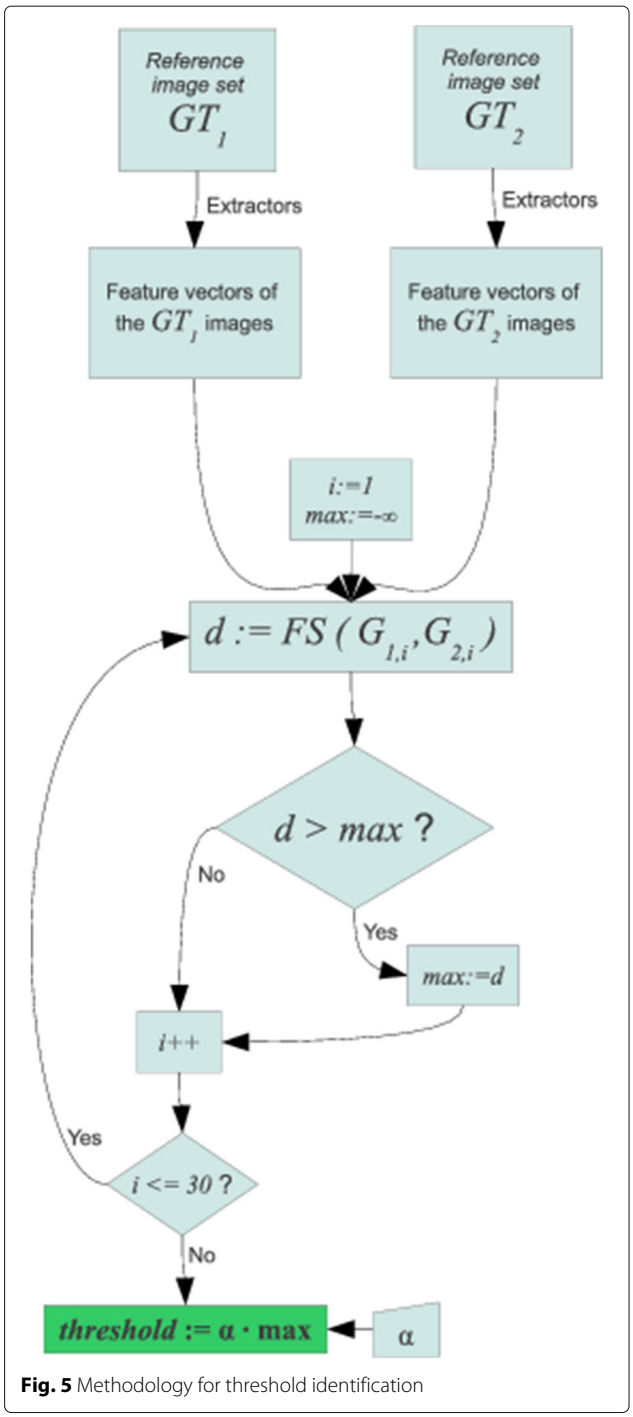

Fig. 5 Methodology for threshold identification

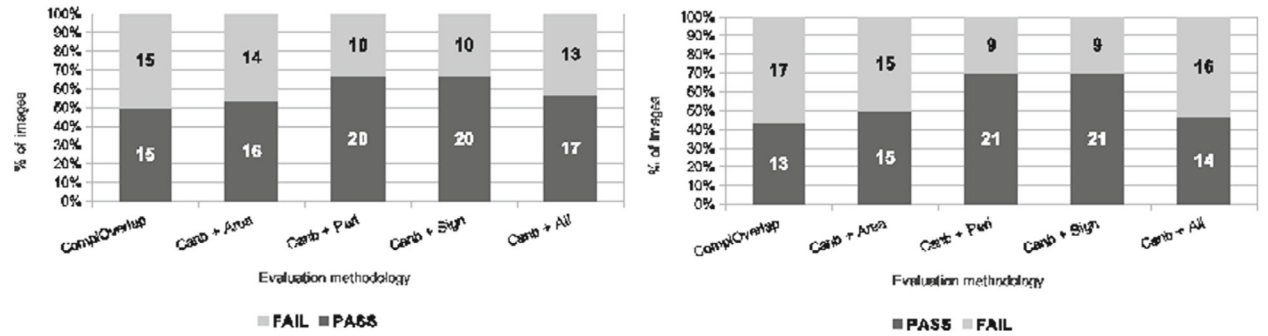

Fig. 6 The approved and disapproved results from the *ComplOverlap* methodology and with the four graphic oracle variations using the *Canberra* distance and thresholds with $\alpha = 1$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

image on the right side. Secondly, a thresholding is executed to transform the original image into a binary one, where the white region represents the breast and the black region is the background. Then, the center point of the right border is selected and, from it, radial lines are drawn from a determined interval of angles, from this point to the first black pixel found (first background pixel). Lastly, all the points found are joined to form the breast edge and all external points to this edge receive the black color.

We used 30 test cases of this scheme in the evaluation experiments. Two individuals manually segmented the breast region in the 30 original mammograms, thus generating two sets of ground truth (GT) composed of the reference images—in the context of our approach—in the test oracles.

For the following definition of the methodologies applied, we defined *Output* as the set of 30 outputs in the segmentation scheme adopted as the program under test.

$$\text{Output} = \{O_1, O_2, O_3, \ldots, O_{30}\}$$

GT_1 and GT_2 are, respectively, the set of reference images produced by individual 1 and the set of reference images produced by individual 2.

$$GT_1 = \{G_{1,1}, G_{1,2}, G_{1,3}, \ldots, G_{1,30}\}$$

$$GT_2 = \{G_{2,1}, G_{2,2}, G_{2,3}, \ldots, G_{2,30}\}$$

Figure 3 shows a mammogram example and its respective segmented images used in the experiments.

The graphic oracles are configured in relation to similarity functions and feature extractors. Therefore, to conduct the case study, we implemented and included different functions and extractors in the O-FIm framework. Each experiment was repeated three times, using a different similarity function in each execution. The results of a previously conducted study [31] determined three major similarity function groups with similar behaviors. For this study, one function from each of these groups was selected.

Similarity functions

We used three different metric distances in the experiments. They are presented as follows, where A and B represent the feature vectors of the two images being compared. In addition, n represents the number of feature extractors used to perform the comparisons.

Canberra distance: the *Canberra* distance between two vectors is given in Eq. 3. The calculation of this distance divides the difference module between each pair of corresponding features by the sum of the feature value module.

$$Dis_{\text{Canb}}(A,B) = \sum_{i=0}^{n-1} \frac{|a_i - b_i|}{|a_i| + |b_i|} \tag{3}$$

Statistical value χ^2: the function determined by the statistical value χ^2, shown in Eq. 4, emphasizes large discrepancies between the feature vectors compared [32]. It also measures how unlikely the vector distribution is when compared to the reference vector [33].

$$Dis_{\chi^2}(A,B) = \sum_{i=0}^{n-1} \frac{(b_i - m_i)^2}{m_i}, \textbf{where } m_i = \frac{a_i + b_i}{2} \tag{4}$$

Euclidean distance: the *Minkowski* or L_p distance is composed of the similarity functions most used in works that include CBIR and other types of content-based retrievals. These are traditionally used distances, but they are often chosen empirically [32]. The general form of distances in this family is shown in Eq. 5.

Table 1 Performances obtained with Canberra distance by combining the two methodologies with the two reference sets (threshold with $\alpha = 1$)

	ComplOverlap (%)	`Canb + All (%)`
Output × GT_1	50	57
Output × GT_2	43	47

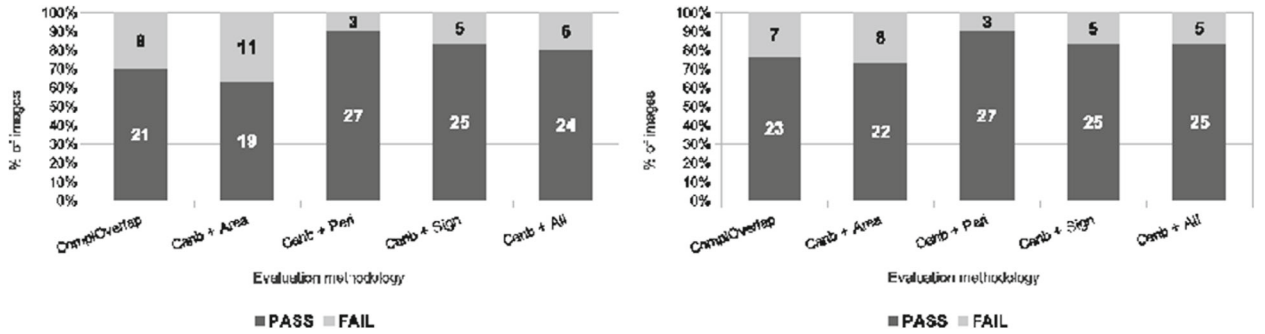

Fig. 7 Approval or disapproval results from the *ComplOverlap* methodology and with the four graphic oracle variations using *Canberra* distance and thresholds with $\alpha = 2$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

$$Dis_{L_p}(A,B) = \sqrt[p]{\sum_{i=0}^{n-1} |a_i - b_i|^p} \tag{5}$$

When $p = 2$, we have the *Euclidean* distance, given in Eq. 6. For this distance, the points whose distances from a reference point (x, y) are less than or equal to a given r value form a circle centered at (x, y) in a two-dimensional space.

$$Dis_{L_2}(A,B) = \sqrt[2]{\sum_{i=0}^{n-1} (a_i - b_i)^2} \tag{6}$$

Feature extractors

Three feature extractors were implemented and included in the O-FIm framework herein. All features were implemented to normalize the computed values in the interval $[0, 1]$.

Area: the *Area* extractor counts the number of pixels that belong to a region of interest in the image (breast area represented). In the images we used, the region of interest is represented by pixels with values greater than zero (in the grayscale), as the images are binarized. The normalizalization of the result is obtained by dividing the number of pixels by the total number of pixels in the image.

Signature: the value computed by the *Signature* extractor represents the breast contour according to its regularity. Therefore, the algorithm finds the center of the breast in the last column of the image, with intervals in degrees, calculating the distance of this center point from the breast contour, and calculates the standard deviation of the values computed (for a perfect circle, this value should be zero). The extractor returns the standard deviation obtained divided by the highest measure calculated in the previous step.

Perimeter: this extractor counts the number of pixels belonging to edge of the region of interest in the image. The value obtained is divided by the total perimeter in the image.

Methodology based on the proposed graphic oracles

To evaluate each output in the segmentation scheme (O_i , $i = 1 \ldots 30$), comparing it with each of its reference images ($G_{1,i}$ e $G_{2,i}$, $i = 1 \ldots 30$), we determined the graphic oracle textually described, as shown next.

```
----------------------------------------
similarity SimilarityFunction
extractor Area { thr = 0 }
extractor Perimeter { thr = 0 }
extractor Signature { thr = 0
angleRadius = 10 }
precision = threshold
----------------------------------------
```

This oracle description considers the features of Area, Perimeter, and Signature (described in the section "Feature extractors") to compose the feature vectors. Thus, the automated segmentation quality can be evaluated by comparing the output vector under test with the vector of its reference image. To do so, a FS similarity function that calculates a measure is used to indicate the difference between vectors. A threshold value indicates the maximum value for this measure, which will indicate that the vectors are sufficiently similar to consider this segmentation correct.

Table 2 Performances obtained with *Canberra* distance by combining the two methodologies with the two reference sets (threshold with $\alpha = 2$)

	ComplOverlap %	`Canb + All %`
Output × GT_1	70	80
Output × GT_2	77	83

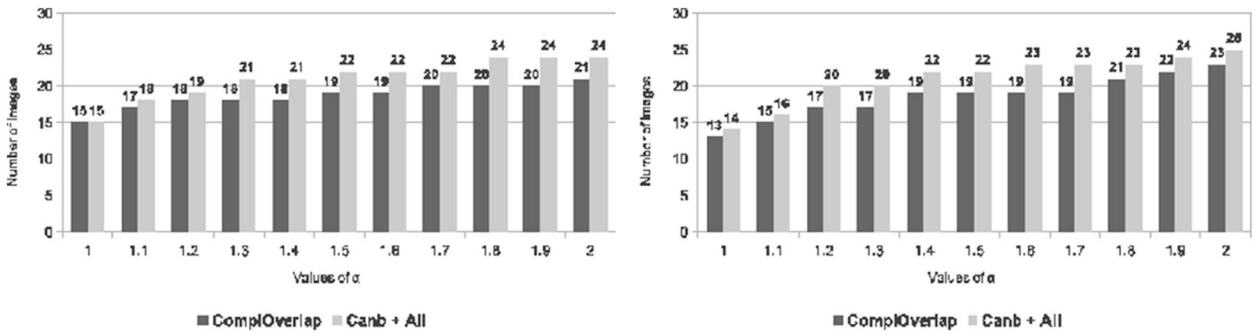

Fig. 8 Approval results from using the *ComplOverlap* methodology and the graphic oracle which used three extractors and the *Canberra* distance, with thresholds with α ranging from 1 to 2. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

All characteristics used in this case study consider all the pixels with zero value in grayscale as background image (meaning of the `thr` parameter that appears in the description of the graphic oracle). The signature extractor uses an interval of 10° (parameter `angleRadius`) to compute the feature value (details are available in the section "Feature extractors"). Note that one of the reasons for choosing these characteristics was their simplicity. This fact demonstrated that extremely complex extractors are not needed to obtain the evaluation results that are consistent with the Overlap measure, which served as a basis for comparison in this study.

Besides the general graphic oracle defined above, comparisons were made using each feature individually. Thus, the contribution and influence of each feature in different test scenarios constructed in this work could be evaluated.

Figure 4 shows a diagram representation for the methodology stages based on the graphic oracles applied herein.

Threshold definition

To establish whether an output tested passed or failed the test (comparison), a threshold value must be determined (threshold) that will indicate the maximum difference that can be computed between two vectors so they can be considered similar.

Given a similarity function $FS(X, Y)$ that calculates the distance between vectors X and Y and any graphic oracle, the approach to determine the threshold value used in this study is shown in the diagram of Fig. 5 and described as follows.

Each image of the reference set GT_1 is compared to its corresponding image in the reference set GT_2, using the oracle under consideration (similarity function and extractors). Thus, the set of distances is obtained, as shown in Eq. 7.

$$Dist(GT_1, GT_2) = \{FS(G_{1,1}, G_{2,1}), FS(G_{1,2}, G_{2,2}), \ldots, FS(G_{1,30}, G_{2,30})\} \quad (7)$$

Fig. 9 Example of segmentation rejected by applying the methodology based on graphic oracles (using the *Canberra* distance, the three extractors, and threshold with $\alpha = 1$), but approved by applying the *ComplOverlap* methodology. The *first* image (O_{27}) is the output image to be evaluated. The *second* ($G_{1,27}$) is the reference image produced by individual 1

Table 3 Value of *ComplOverlap* and *Canberra* distances calculated from the images in Fig. 9 and their respective evaluation results

	ComplOverlap	Area	Perimeter	Signature	All
Value calculated [a]	0.020	0.009	0.027	0.121	0.157
threshold	0.033	0.011	0.036	0.056	0.090
Test result	PASS	PASS	PASS	FAIL	FAIL

[a] *ComplOverlap* and distances

The threshold value can then be calculated using Eq. 8.

$$\text{threshold} = \alpha \cdot \max\left(Dist\left(GT_1, GT_2\right)\right) \quad (8)$$

As the distances between the images of GT_1 and GT_2 are small—as they were manually processed—the α value represents the number of times the distance between an output and its reference image can be greater than the maximum distance between two reference images of the sets GT_1 and GT_2 for the same output. In this study, we conducted experiments using different values for α, varying them from 1 to 2 at fixed intervals.

Methodology based on the Overlap measure

The Overlap measure was determined in Eq. 1. To compare the evaluation results from the methodology based on graphic oracles and the results from the Overlap measure, this measurement had to be adapted. The first step was to take the complement of its value, that is, subtract the computed value of 1, as shown in Eq. 9.

$$\text{ComplOverlap}\left(A_{\text{seg}}, A_{\text{man}}\right) = 1 - \frac{|A_{\text{seg}} \cap A_{\text{man}}|}{|A_{\text{seg}} \cup A_{\text{man}}|} \quad (9)$$

Thus, the closer to zero the ComplOverlap value is, the better the segmentation performance. This measurement transformation was performed to allow calculating a threshold similar to that shown in the Section "Threshold definition".

The ComplOverlap values were calculated from the image pairs corresponding to sets GT_1 and GT_2, obtaining set V (Eq. 10).

$$V\left(GT_1, GT_2\right) = \left\{\text{ComplOverlap}\left(G_{1,1}, G_{2,1}\right), \ldots, \text{ComplOverlap}\left(G_{1,30}, G_{2,30}\right)\right\} \quad (10)$$

Similar to that shown in the Section "Threshold definition," the threshold value was determined by using Eq. 11.

$$\text{threshold} = \alpha \cdot \max\left(V\left(GT_1, GT_2\right)\right) \quad (11)$$

The ComplOverlap value was calculated to evaluate each output of the segmentation scheme with its respective reference images of sets GT_1 and GT_2. The threshold values determined were then used to evaluate whether the output passed different test scenarios (possible due to the change in value α). For automating these procedures, algorithms were implemented to calculate the ComplOverlap between a pair of images (representing the segmented regions) and to evaluate the results.

A coherent comparison was thus made between the methodology based on graphic oracles and the methodology based on Overlap measure, the ComplOverlap.

Results and discussion

The performance of the segmentation scheme was determined as a quality criterion to evaluate the number of images approved, that is, those which, according to the evaluation method used, were correctly segmented by the system. The performance measure considered was the percentage of approved images.

All the experiments that included the defined graphic oracles were performed three times. A different similarity function was used in each run (see the Section "Similarity functions").

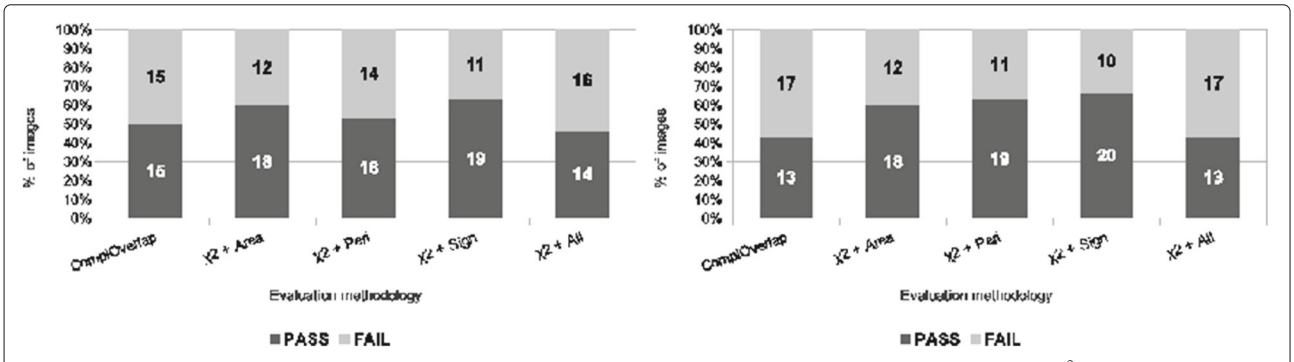

Fig. 10 Approval or disapproval results from the *ComplOverlap* methodology and the four graphic oracle variations using χ^2 function and thresholds with $\alpha = 1$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

Table 4 Performance obtained with *statistical value* χ^2 by combining the two methodologies with two sets of reference (threshold with $\alpha = 1$)

	ComplOverlap (%)	χ^2+ All (%)
Output × GT_1	50	47
Output × GT_2	43	43

Results from the Canberra distance

The graphs in Fig. 6 show the performance of the segmentation scheme evaluated according to the two methodologies described: the method based on the Overlap measurement (ComplOverlap) and the four variants in the methodology based on graphic oracles (using each extractor individually and the three together). We considered $\alpha = 1$ to determine the thresholds.

Comparing the number of approved and disapproved images, for the set of reference images GT_1, the ComplOverlap methodology was verified to have approved two images less than the "`Canb + All`" oracle. Using the set GT_2, only one image was. Therefore, when the two methodologies are compared, no significant variation was observed in the number of approved images, given that the greatest difference was of only two images. In this experiment, the ComplOverlap method proved to be more rigorous for evaluating the images.

Applying each extractor individually for both sets of reference, a less significant variation was obtained in the number of images approved for the Area extractor (`"Canb + Area"`)—one more image was approved for GT_1 and two more images for GT_2. More significant variations, that is, a greater number of approved images was observed in the Perimeter (`"Canb + Peri"`) and Signature (`"Canb + Sign"`) extractors—five more images approved for GT_1 and eight more images for GT_2.

According to these results, for the *Canberra* distance, the most critical feature (the one that most disapproved) was the Area, which significantly influenced the results when all three extractors were applied in unison. Note how in this result, in the graphic oracle approach, each extractor is individually important in the comparison. This means that the evaluator or system tester can set the characteristics that are important for a satisfactory segmentation and use them in the evaluation. In fact, the empirical results showed that, excluding only the Area extractor, the number of approved images increased to 21 when the reference set GT_1 was used, and increased to 19 when GT_2 was used.

The Overlap measure evaluates only the features of the area in the segmentation and the location of the segmented region. Thus, the quality of the system under evaluation is intrinsically related just to these characteristics. In the methodology based on graphic oracles, many other characteristics may be incorporated, such as the regularity and circularity of the edge, and other features, hence enriching the test process and increasing the tester flexibility.

In this first experiment, in relation to ComplOverlap and also to the three characteristics applied together, the performance of the system changed, as shown in Table 1. For both sets of reference, the performances remained in the same range—between 50 and 60% of the images approved for GT_1 and between 40 and 50% of approved images for GT_2—divergence not exceeding 10%.

The evaluation results considering the value of α equal to 2 are shown in the graphs of Fig. 7. By multiplying the thresholds by a factor of 2, both assessment methods proved to be more flexible, approving a higher number of outputs. The variation between the ComplOverlap methodology and the graphic oracle methodology increased when the Canberra distance and the three extractors were applied. ComplOverlap approved three images less using GT_1, and two images less using GT_2.

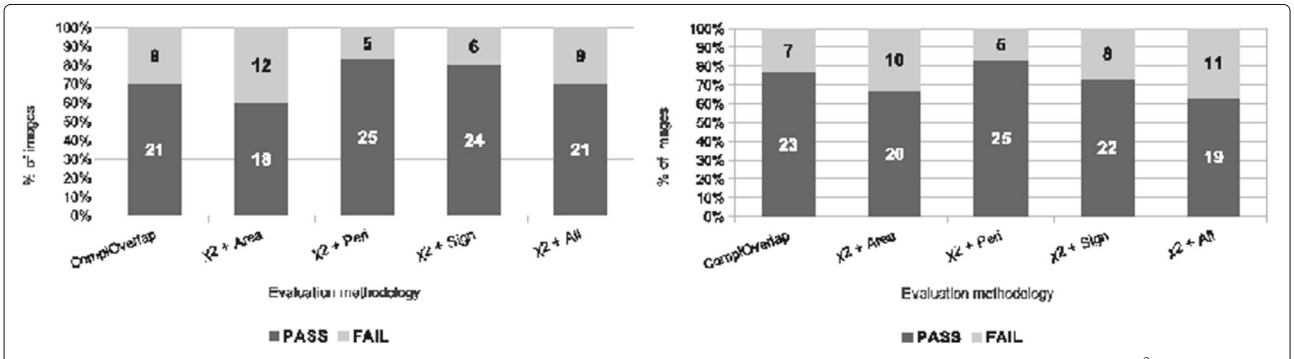

Fig. 11 Approval or disapproval results from the *ComplOverlap* methodology and the four graphic oracle variations using function χ^2 and thresholds with $\alpha = 2$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

Table 5 Performances obtained with the *statistical value* χ^2 by combining the two methodologies with two sets of reference (threshold with $\alpha = 2$)

	ComplOverlap (%)	χ^2+ All (%)
Output × GT_1	70	70
Output × GT_2	77	63

However, the consistency between the two methodologies is still verifiable with these results, given that, again, the divergence between the methods was small—three and two images, respectively.

Table 2 shows the performances of the segmentation scheme obtained for ComplOverlap and the graphic oracle with three extractors, applying thresholds with $\alpha = 2$. Table 2 highlights the major differences mentioned in the results of the two methods, however with a difference not as great as that observed for $\alpha = 1$. Also in this experiment, the difference in percentage of approved images did not exceed 10% for both reference sets.

Next, the α values varied between 1 and 2, taking intervals of 0.1 in the ComplOverlap methodology, as well as in the graphic oracle methodology that applied the three extractors together. The graphs of Fig. 8 show the results from this variation. Overall, for both sets of reference, the difference between the number of approved images, using both methods, was observed not to exceed 13% (four images) for any α value. For most of the α values, this percentage was even lower. This result demonstrates the consistency between methodologies and shows that both had similar performances for the system tested under most of the conditions used, especially for lower α values.

Also important is the fact that for all of the α values used, the ComplOverlap methodology was more rigorous in the evaluation of the segmentation system, that is, fewer images were approved. This shows that the inclusion of characteristics the Overlap measure does not consider, such as Signature, influences the evaluation of the results.

Moreover, the results showed that the same images approved in the ComplOverlap methodology were not always also approved in the graphic oracle methodology or vice versa. This result can be explained taking into consideration that in the graphic oracle methodology, different characteristics were considered and, as already mentioned, each one had its individual influence in the comparison. Thus, the most significant change of a given characteristic may cause an image to be rejected by the graphic oracle, but that the ComplOverlap measure does not vary so much as to reject the same image. Figure 9 shows an example.

Between the output image evaluated and its reference image in the GT_1 set, there is a uniformity discrepancy on the edge of the segmented region, especially the top and bottom extremities. In fact, as shown in Table 3, the Signature extractor reflected this discrepancy by using the Canberra distance when comparing the image vectors. Individually, this was the only extractor responsible for rejecting the image in the methodology based on graphic oracles. As it has no control over the characteristics related to the edge uniformity, the ComplOverlap methodology was not able to reflect the discrepancy in the images.

Results from statistical value χ^2

The graphs of Fig. 10 show the results of experiments that applied the statistical value χ^2 as similarity function in the graphic oracles determined. Results from the ComplOverlap methodology in these graphs are the same as in Fig. 6, as there are no versatile parameters in this methodology, except for the α value to define the threshold.

For both reference sets, with similarity function χ^2, the performances obtained with ComplOverlap and with the graphic oracle that used the three extractors ("χ^2 +All") were verified to be basically the same, varying in only one image, the reference set GT_1. Table 4 shows that these performance values varied within the same range as

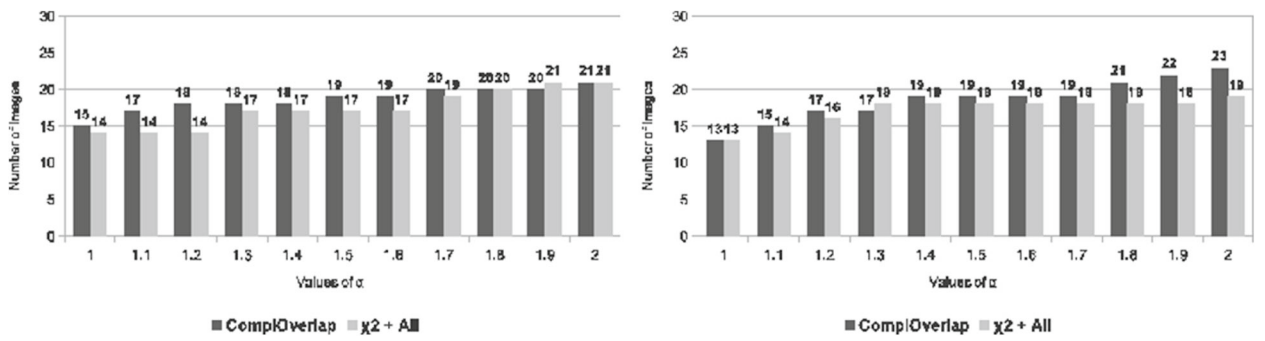

Fig. 12 Approval results from the *ComplOverlap* methodology and with the graphic oracle that used the three extractors and the function χ^2, using the thresholds with α ranging from 1 to 2. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

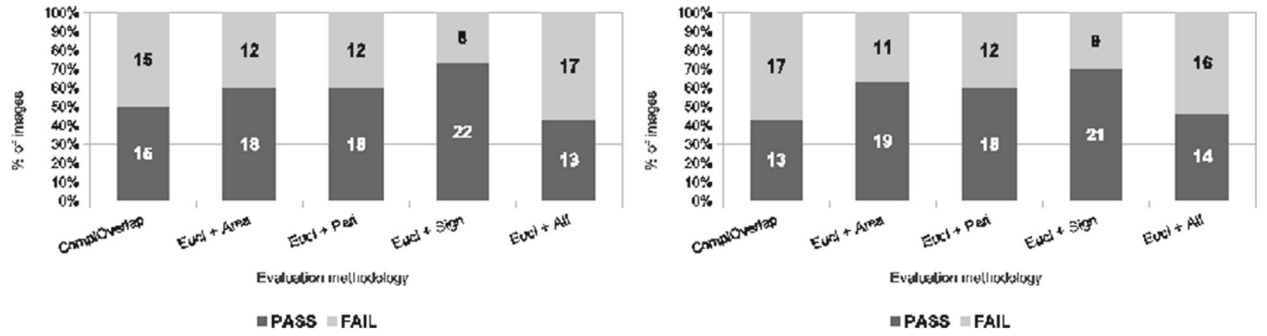

Fig. 13 Approval or disapproval results from the *ComplOverlap* methodology and the four graphic oracle variations using *Euclidean* distance and thresholds with $\alpha = 1$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

the values observed, under the same conditions using the Canberra distance. However, no performance over 50% was achieved.

For the extractors applied individually with the function χ^2, for $\alpha = 1$, no characteristic was significantly more critical than the others. This result could explain the lower variation observed for the results of the ComplOverlap methodology and the results of the graphic oracle using the three extractors together.

Figure 11 shows the results achieved for thresholds with $\alpha = 2$. Using the reference set GT_1, the performance achieved with ComplOverlap was similar to that observed using the graphic oracle, which used the three extractors. As for the reference set GT_2, the methodology ComplOverlap approved four more images. Table 5 shows the performances obtained.

Individually applying the extractors to the graphic oracle approach showed that the increase in thresholds highlighted the Area feature as the one that least approved outputs for both sets of reference, unlike the results from $\alpha = 1$, in which this difference was less significant.

Figure 12 shows the results from varying the values of α from 1 to 2. Also for the χ^2 function, the difference between the number of approved images did not exceed 13% for any value of α when comparing the two methods. For most of the α values, this difference is of only one or two image outputs. These graphs also show the thresholds used to obtain similar performances in the segmentation system.

Unlike the Canberra distance, similarity function χ^2 was more rigorous in most thresholds than the ComplOverlap methodology.

Results from the Euclidean distance

Figure 13 shows that by using the Euclidean distance, together with $\alpha = 1$ thresholds, our methodology again showed similar performances to those of ComplOverlap. The results varied in only two outputs when the reference set GT_1 was used and one output when GT_2 was used.

Individually applying each extractor in the graphic oracle methodology, higher numbers of approved outputs are observed, with some performance variation between extractors. Thus, it is important to check the consistency of the results from the ComplOverlap methodology and the results from the graphic oracles, regardless of the similarity function used.

Table 6 shows the performances obtained with Euclidean distance, using thresholds with $\alpha = 1$, compared to the performances obtained with the ComplOverlap methodology (those already compared with the performances of the two other similarity functions). These results indicated similar performance to that of the other two similarity functions applied.

Figure 14 shows the results from using thresholds with $\alpha = 2$. The greatest difference between the results from the ComplOverlap methodology and the results from graphic oracles was observed when the images of the GT_1 set were used as references for the outputs tested. In this situation, ComplOverlap approved four images less compared with the oracle, which used the three extractors together.

Even so, the performances observed for this threshold value, with the Euclidean distance, were consistent with the performances that used the Canberra distance. Table 7 shows the performances achieved with the Euclidean distance together with the performances achieved with the ComplOverlap methodology.

Table 6 Performances obtained using *Euclidean* distance by combining the two methodologies with the two sets of reference (threshold with $\alpha = 1$)

	ComplOverlap (%)	`Eucl + All` (%)
Output × GT_1	50	43
Output × GT_2	43	47

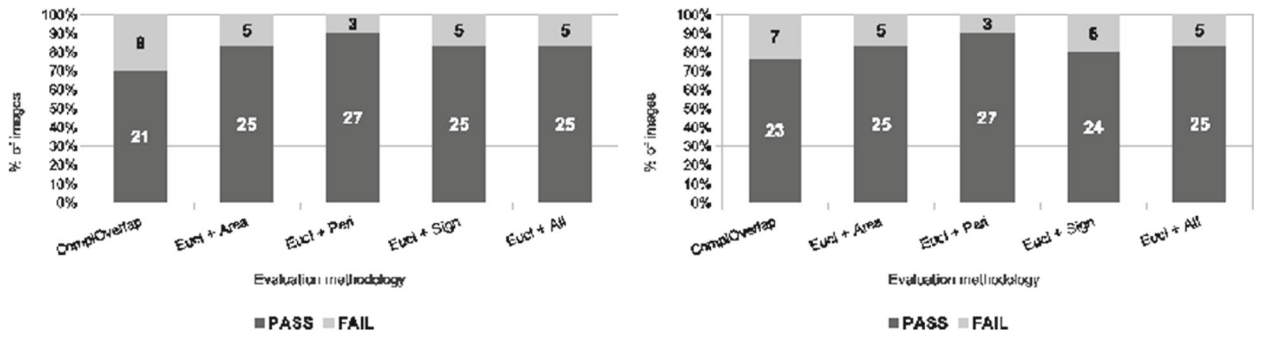

Fig. 14 Approval or disapproval results from the *ComplOverlap* methodology and the four graphic oracle variations using the *Euclidean* distance and thresholds with $\alpha = 2$. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

Again, considering the extractors individually, there were variations in the number of approved outputs comparing extractor against extractor. Even with these differences, comparing the results of the four oracles and the ComplOverlap methodology, it was verified that the performance of the segmentation schemes did not significantly differ from each other.

Figure 15 shows the α variation between 1 and 2, the approval results for the outputs of ComplOverlap and of the oracle of three extractors. Again, the difference between the two methodologies did not exceed 13% of the images approved for any value of α. There were smaller variations for higher values of α and for the set GT_1 also lower values of α. There were also thresholds that led to equivalent performances using both sets of references.

Considering the reference set GT_1, the ComplOverlap was more rigorous than the graphic oracle only after value α =1.5. Considering the set GT_2, for any value of α tested, ComplOverlap was more rigorous or produced results equivalent to those with the graphic oracle.

Comparison between the similarity functions used

With the results presented in the previous subsections, the performances obtained were in accordance with the results of the three similarity functions used. Differences were observed for these results, which were expected, since functions with behaviors different from each other were selected, according to a previous work on similarity functions [31].

The graphs in Fig. 16 show a comparison between the results from using each of the similarity functions and from the ComplOverlap methodology.

It was observed that for most α values tested, the Euclidean and Canberra distances were less rigorous than the ComplOverlap methodology. However, function χ^2 was more rigorous than this metric in most cases.

Characteristics, advantages, and limitations of the methodology based on graphic oracles

The main limitation or disadvantage of the proposed methodology regards the computational cost involved. The more images to be tested, the more feature vectors to be computed. The greater the vector size, the more processing is required. This disadvantage can be minimized by optimizing the algorithms implemented with effective indexing techniques (for example, when there is a fixed set of images it is not necessary to always calculate the same features if their values are efficiently stored) and with size reduction techniques for the vectors.

Even with this limitation, the advantage of adapting the methodology to the evaluation criteria for each particular system, determining the specific extractors, is very interesting and powerful. The experiments conducted demonstrated the consistency of the results with results provided by a technique that is widely used in the literature. However, although the methodology based on the Overlap measure, in the specific case of the experiments discussed here, exhibited similar results, it does not have the same flexibility as the proposed methodology.

Although apparently simpler, the intersection between the images compared does not allow finding differences that could lead to improving the algorithms implemented in a CAD scheme. However, using the specific extractors, as proposed in this study, can help to more precisely identify defects in the software. For example, the preprocessing algorithms may be distorting the edges of the

Table 7 Performances obtained using *Euclidean* distance by combining the two methodologies with the two sets of reference (threshold with $\alpha = 2$)

	ComplOverlap (%)	`Eucl + All` (%)
Output × GT_1	70	83
Output × GT_2	77	83

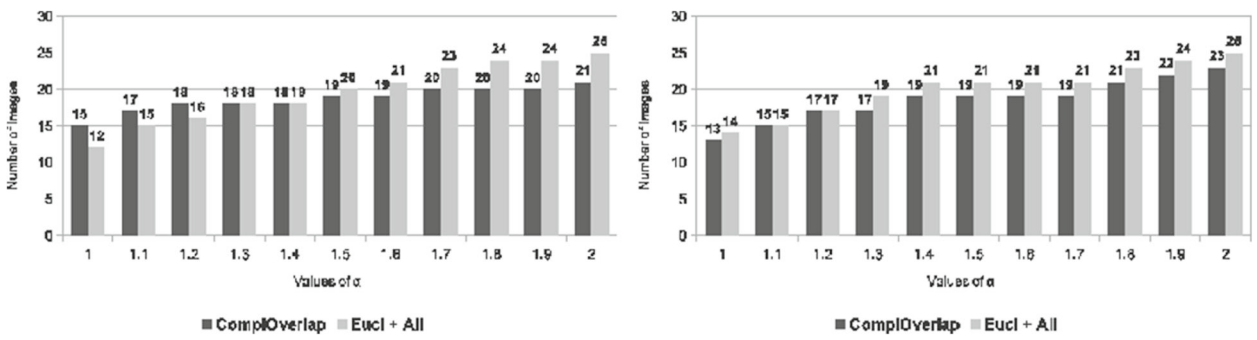

Fig. 15 Approval or disapproval results from the *ComplOverlap* methodology and with the graphic oracle that used the three extractors and the *Euclidean* distance, using thresholds ranging α from 1 to 2. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

structures of interest, a defect that an edge extractor can more clearly point to, as is the case of the example shown in Fig. 9.

With all the results shown, the methodology based on graphic oracles was concluded to be a robust tool to evaluate the performance of segmentation schemes. A key attribute of this methodology is that the evaluator can define which criteria are deemed important for evaluating the system and transform them into feature extractors. Thus, during the tests, only the essential can be taken into consideration. For example, if the regularity of the edge of the segmented region is not so relevant to consider whether the segmentation is accurate or not, a Signature extractor will not be necessary. However, if the size and location of one or more objects in the image are relevant, then the Area and Center of Mass extractors [1], for example, may be effective for composing feature vectors.

Furthermore, the O-FIm framework is a free tool that is available and can be used to configure graphic oracles, as well as serve as API for implementing test scripts that use such oracles to evaluate systems with graphical outputs.

Conclusions

This paper presented the results of a case study in which an evaluation methodology was tested on a segmentation scheme for mammographic images. The proposed methodology is based on graphic oracles and uses the O-FIm framework as a tool for configuring the oracles and for conducting the tests.

The contribution of this paper is to provide a flexible methodology for evaluating segmentation schemes, which can include features of interest, besides the segmented area. The O-Fim framework is distributed as free software and available for public access. The feature extractors used in the experiments were also available and can be easily reused. In addition, other extractors can be implemented, including extractors based on techniques used in the segmentation process itself.

The results in this work demonstrate the validity of the proposed methodology and its consistency with the results of a second methodology based on the Overlap measure, a metric that has been used in many works found in the literature to evaluate segmentation schemes. In the experiments conducted, the proposed methodology proved to be robust for the similarity function used

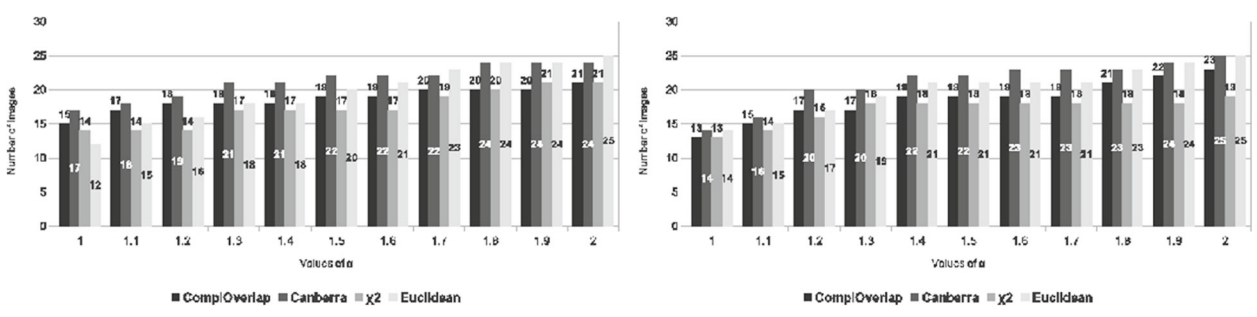

Fig. 16 Comparison of the approval results from using the *ComplOverlap* methodology and each one of the similarity functions applied. The *first* diagram shows the evaluation results using as references the images manually segmented by individual 1 (GT_1). The *second* diagram used as references the images manually segmented by individual 2 (GT_2)

and also flexible and adaptable to effectively evaluate segmentation schemes.

Endnote

[1] http://www2.ccsl.icmc.usp.br/pt-br/projects/o-fim-oracle-images

Acknowledgements
The authors would like to thank The State of São Paulo Research Foundation (Fundação de Amparo à Pesquisa do Estado de São Paulo) (Fapesp)—Process 2010/15691-0 and the Brazilian National Council of Scientific and Technological Development (Conselho Nacional de Desenvolvimento Cientìfico e Tecnológico) (CNPq)—Processes 559931/2010-7 and 401745/2013-9, and the National Institute of Science and Technology—Medicine Assisted by Scientific Computing (Instituto Nacional de Ciência e Tecnologia—Medicina Assistida por Computação Científica)—INCT-MACC.

Authors' contributions
VMG contributed in the implementation of the features and similarity functions, experimental study planning and execution, and text writing. MD contributed in the framework conception and implementation, planning and evaluation of the experimental studies, and text writing. FN was involved in the conception of the project idea, conception of the case studies, definition of the features to be used in the experimental studies, and text writing. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Laboratory of Computer Applications for Health Care, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, Rua Arlindo Bettio, 1000, Vila Guaraciaba, 03828-000 São Paulo, Brazil. [2] Laboratory of Software Engineering, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Avenida Trabalhador Sancarlense, 400, 13566-590 São Carlos, Brazil. [3] Escola Politécnica, Universidade de São Paulo, Avenida Professor Luciano Gualberto, travessa 3, 380, Butantã, 05508-010 São Paulo, Brazil.

Received: 7 July 2015 Accepted: 22 December 2016
Published online: 19 January 2017

References

1. Gonzalez RC, Woods RE (2008) Digital Image Processing. 3rd edn. Pearson Education, New Jersey
2. Zheng B, Pu J, Park SC, Zuley M, Gur D (2008) Medical Imaging 2008: Computer-Aided Diagnosis. In: Giger ML, Karssemeijer N (eds). Medical Imaging 2008: Computer-Aided Diagnosis. SPIE Vol. 6915. pp 691530–169153011. http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=2008SPIE.6915E....G&data_type=BIBTEX&db_key=PHY&nocookieset=1
3. Bastos CACM, Tsang IR, Vasconcelos GS, Cavalcanti GDC (2012) Pupil segmentation using pulling and pushing and BSOM neural network. In: Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference On. pp 2359–2364. doi:10.1109/ICSMC.2012.6378095
4. Banerjee B, Surender VG, Buddhiraju KM (2012) Satellite image segmentation: a novel adaptive mean-shift clustering based approach. In: Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International. pp 4319–4322. doi:10.1109/IGARSS.2012.6351712
5. Deepa P, Geethalakshmi SN (2011) Improved watershed segmentation for apple fruit grading. In: Process Automation, Control and Computing (PACC), 2011 International Conference On. pp 1–5. doi:10.1109/PACC.2011.5979003
6. Huddar SR, Gowri S, Keerthana K, Vasanthi S, Rupanagudi SR (2012) Novel algorithm for segmentation and automatic identification of pests on plants using image processing. In: Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference On. pp 1–5. doi:10.1109/ICCCNT.2012.6396012
7. Gonçalves VM, Delamaro ME, Nunes FLS (2014) A systematic review on evaluation and characteristics of computer-aided diagnosis systems. Rev Bras Engenharia Biomédica 30(4):355–383
8. Gruszauskas NP, Drukker K, Giger ML, Sennett CA, Pesce LL (2008) Performance of breast ultrasound computer-aided diagnosis: dependence on image selection. Acad Radiol 15(10):1234–1245
9. Korfiatis P, Skiadopoulos S, Sakellaropoulos P, Kalogeropoulou C, Costaridou L (2007) Automated 3D segmentation of lung fields in thin slice CT exploiting wavelet preprocessing. In: Proceedings of the 12th International Conference on Computer Analysis of Images and Patterns. Springer, Berlin. pp 237–244
10. Odet C, Belaroussi B, Benoit-Cattin H (2002) Scalable discrepancy measures for segmentation evaluation. In: Image Processing. 2002. Proceedings. 2002 International Conference On Vol. 1. pp 785–7881. doi:10.1109/ICIP.2002.1038142
11. Goumeidane AB, Khamadja M, Belaroussi B, Benoit-Cattin H, Odet C (2003) New discrepancy measures for segmentation evaluation. In: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference On Vol. 2. pp 411–143. doi:10.1109/ICIP.2003.1246704
12. Gelasca ED, Ebrahimi T, Farias MCQ, Carli M, Mitra SK (2004) Towards perceptually driven segmentation evaluation metrics. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference On. pp 52–52. doi:10.1109/CVPR.2004.191
13. Hachouf F, Ahmed Seghir Z, Zeggari A (2006) A generic methodology for image segmentation evaluation. In: Information and Communication Technologies, 2006. ICTTA '06. 2nd Vol. 1. pp 1794–1799. doi:10.1109/ICTTA.2006.1684658
14. Cardoso JS, Corte-Real L (2005) Toward a generic evaluation of image segmentation. IEEE Trans Image Process 14(11):1773–1782. doi:10.1109/TIP.2005.854491
15. Peng B, Li T (2013) A probabilistic measure for quantitative evaluation of image segmentation. Signal Process Lett, IEEE 20(7):689–692. doi:10.1109/LSP.2013.2262938
16. Delamaro ME, Nunes FLS, Oliveira RAP (2011) Using concepts of content-based image retrieval to implement graphical testing oracles. Softw Test Verif Reliab 23(3):171–198
17. Nunes FLS, Schiabel H, Goes C (2007) Contrast enhancement in dense breast images to aid clustered microcalcifications detection. J Digit Imaging 20(1):53–66
18. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):5–1560
19. El-Naqa I, Yang Y, Galatsanos NP, Nishikawa RM, Wernick MN (2004) A similarity learning approach to content-based image retrieval: application to digital mammography. IEEE Trans Med Imaging 23(10):1233–1244. doi:10.1109/TMI.2004.834601
20. Traina Jr C, Traina A, Faloutsos C, Seeger B (2002) Fast indexing and visualization of metric data sets using slim-trees. IEEE Trans Knowl Data Eng 14(2):244–260
21. Petrakis EGM, Faloutsos C, Lin KI (2002) Imagemap: an image indexing method based on spatial similarity. IEEE Trans Knowl Data Eng 14(5):979–987
22. Baresi L, Young M (2001) Test oracles. Technical Report CIS-TR-01-02, University of Oregon, Dept. of Computer and Information Science, Eugene, Oregon, USA. https://people.eecs.ku.edu/~saiedian/Teaching/Fa07/814/Resources/oracles.pdf
23. Hoffman D (1998) A taxonomy for test oracles. In: Proceedings of the 11th International Quality Week. Software Research Institute, Inc, San Francisco. pp 1–8
24. Schilham AMR, van Ginneken B, Loog M (2006) A computer-aided diagnosis system for detection of lung nodules in chest radiographs with an evaluation on a public database. Med Image Anal 10(2):247–258
25. Song E, Xu S, Xu X, Zeng J, Lan Y, Zhang S, Hung CC (2010) Hybrid segmentation of mass in mammograms using template matching and dynamic programming. Acad Radiol 17(11):1414–1424
26. Tan NM, Liu J, Wong DWK, Yin F, Lim JH, Wong TY (2010) Mixture model-based approach for optic cup segmentation. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE. pp 4817–4820
27. Huang SF, Chaoa HY, Hsu CC, Yang SF, Kao PF (2009) A computer-aided diagnosis system for whole body bone scan using single photon emission computed tomography. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. IEEE. pp 542–545
28. Jiménez S, Alemany P, Fondón I, Foncubierta A, Acha B, Serrano C (2010) Detección automática de vasos en retinografías. Arch Soc Esp Oftalmol 85:103–109

29. Pietka E, Kawa J, Badura P, Spinczyk D (2010) Open architecture computer-aided diagnosis system. Expert Syst 27(1):17–39
30. Garnavi R, Aldeen M, Celebi ME (2011) Weighted performance index for objective evaluation of border detection methods in dermoscopy images. Skin Res Technol 17(1):35–44
31. Nunes FLS, Delamaro ME, Gonçalves VM, Lauretto MS (2015) CBIR based testing oracles: an experimental evaluation of similarity functions. Int J Softw Eng Knowl Eng 25(08):1271–1306
32. Bugatti PH, Traina AJM, Traina-Jr C (2008) Assessing the best integration between distance-function and image-feature to answer similarity queries. In: Proceedings of the 2008 ACM Symposium on Applied Computing. SAC '08. ACM, New York. pp 1225–1230
33. Ponciano-Silva M, Traina AJM, Azevedo-Marques PM, Felipe JC, Traina-Jr C (2009) Including the perceptual parameter to tune the retrieval ability of pulmonary CBIR systems. In: Proceedings of the 22nd IEEE International Symposium on Computer-Based Medical Systems. IEEE, Albuquerque. pp 1–8