Optimization of Computer Aided Detection systems: an evolutionary approach

Optimization of Computer Aided Detection systems: an evolutionary approach

Lia Morra[a,b,*], Nunzia Coccia[b], Tania Cerquitelli[b]

[a] *im3D Turin, Italy*
[b] *Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy*

Abstract

Computer Aided Diagnosis (CAD) systems are designed to aid the radiologist in interpreting medical images. They are usually based on lesion detection and segmentation algorithms whose performance depends on a large number of parameters. While time consuming and sub-optimal, manual adjustment is still widely used to adjust parameter values. Genetic or evolutionary algorithms (GA) are effective optimization methods that mimic biological evolution. Genetic algorithms have been shown to efficiently manage complex search spaces, and can be applied to all kinds of objective functions, including discontinuous, non-differentiable, or highly nonlinear ones. In this study, we have adopted an evolutionary approach to the problem of parameter optimization. We show that the genetic algorithm is able to effectively converge to a better solution than manual optimization on a case study for digital breast tomosynthesis CAD. Parameter optimization was framed as a constrained optimization problem, where the function to be maximized was defined as weighted sum of sensitivity, false positive rate and segmentation accuracy. A modified Dice coefficient was defined to assess the segmentation quality of individual lesions. Finally, all viable solutions evaluated by the GA were studied by means of exploratory data analysis techniques, such as association rules, to gain useful insight on the strength of the influence of each parameter on overall algorithm performance. We showed that this combination was able to identify multiple ranges of viable solutions with good segmentation accuracy.

Keywords: computer aided detection, genetic algorithms, breast tomosynthesis, segmentation, optimization

1. Introduction

Computer Aided Detection/Diagnosis (CAD) systems are designed to aid the radiologist in interpreting medical images. The acronym CAD denotes a wide variety of systems, whose purpose can be either (i) to identify potential lesions and bring them to the attention of

*I am corresponding author
Email addresses: liamorra@gmail.com (Lia Morra), coccianunzia@gmail.com (Nunzia Coccia), tania.cerquitelli@polito.it (Tania Cerquitelli)

the radiologist (Computer Aided Detection or CADe), or (ii) to estimate the likelihood that imaging features represent an actual disease process (Computer Aided Diagnosis or CADx) (Li and Nishikawa, 2015; Doi, 2007). CAD has also been shown to reduce reading time and reduce variability among readers, especially when radiologists workload is high, disease prevalence is low, and datasets have increased in size and complexity with advances in technology such as Digital Breast Tomosynthesis, CT Colonography or lung CT scans (Morra et al., 2015; Delsanto et al., 2008). In breast screening with digital breast tomosynthesis, there are usually 5 to 7 cancer cases every 1000 women (Houssami, 2015).

Many CAD applications are implemented through a pipeline of image processing and pattern recognition modules. First initial candidates are identified by applying pattern recognition techniques, yielding a list of potential abnormalities or lesion candidates. A segmentation step may be added to obtain a more precise outline of each potential abnormality. Finally, a classifier is applied to reduce the number of false positive candidates based on feature extracted from the candidate. CADx systems will also include a classification step designed to provide the clinician with an assessment of disease, disease type, severity, stage, progression or regression (Doi, 2007). In this paper, we will focus on the detection and segmentation pipeline in which candidate lesions are selected, and then the location and contours of candidate lesions are extracted. Such algorithms rely on a series of parameters that need to be optimized for a specific application. Segmentation algorithms often exhibit complex parameter search spaces with mutually dependent parameters. Furthermore, large variations in image quality and appearances may occur due to differences in acquisition, reconstruction or post-processing between different vendors and/or acquisition protocols, and require algorithm retraining to achieve optimal performance. Deep learning approaches, that are rapidly becoming the state of the art for many applications, also require the manual optimization of several hyper-parameters (Greenspan et al., 2016; Albelwi and Mahmood, 2016). Self-optimization of machine learning pipelines is a common problem in computer vision, as well as in the general machine learning and data analytics community (Cerquitelli et al., 2016; Corso et al., 2017).

Manual tuning of hyper-parameters by trial and error is time consuming and may easily yield sub-optimal results. It is therefore very important to identify algorithmic approaches to efficiently and effectively adapting new or existing algorithms for specific applications, or to tune existing algorithms to different acquisition parameters (Jrgen and Cristian, 2016).

Optimization of detection and segmentation pipelines for CAD applications require the simultaneous optimization of several goals: generally speaking, we aim at high sensitivity, high specificity (i.e. low number of false positives) and high segmentation quality. The main contribution of this paper is to define an optimization framework for detection and segmentation pipeline, based on genetic algorithms, that allows to explore its parameter / hyper-parameter space and optimize the solution with respect to sensitivity, specificity and quality of segmentation. As case study, a simple algorithm for mass detection in Digital Breast Tomosynthesis (DBT) was selected.

Genetic algorithms (GAs) are effective optimization methods based on a natural selection process that mimics biological evolution. GAs have been shown to efficiently manage complex search spaces, and can be applied to all kinds of objective functions, including dis-

continuous, non–differentiable, stochastic, or highly nonlinear ones: the only requirement is that, for any given individual, the value of the objective function can be calculated (Delsanto et al., 2006).

The main strength of GAs is their ability to simultaneously assess a large number of individuals by computing their objective function; their inherent parallelism allows efficient exploitation of parallel and/or distributed architectures. However, when calculating the objective function is not a trivial task, the optimization process can quickly become very computationally intensive, thus limiting the applicability of GAs to large scale problems. This issue can be reduced by careful engineering, such as using proper parameter encoding to optimize the search space.

On the other hand, GAs exploration of the parameter space may yield qualitative and quantitative information on the relative impact of each parameter on overall performance, by examining all individuals found in all generations (Delsanto et al., 2004). To this end, we tracked all viable solutions to the segmentation problem found at each generation, and used associative rules to infer the effect of each parameter on segmentation quality.

The paper is organized as follows: related work are discussed in Section 2, while in Section 3, the proposed evolutionary optimization framework is described, and a specific case study (mass detection and segmentation in digital breast tomosynthesis) is presented. In Section 4, the experimental setup and data analysis techniques are introduced. In Sections 5 and 6, results are presented and discussed.

2. Related work

As stated in the introduction, CAD applications are usually implemented through several steps of image processing and machine learning techniques, including candidate detection, segmentation, feature extraction and classification. While there is an extensive body of literature on optimal feature selection (Sahiner et al., 2000, 2002; Giannini et al., 2013), parameters of image processing pipelines are often manually selected by trial and error, until accurate results on a representative dataset are reached. Nonetheless, a few authors have tackled this issue both in the medical imaging (Anastasio et al., 1998; Angelie et al., 2005; Nemoto et al., 2017; Teare et al., 2017; Wang et al., 2017), bioinformatics (Held et al., 2013; Teodoro et al., 2016) and computer vision communities (Miikkulainen et al., 2017; Stanley and Miikkulainen, 2002). In most cases, the authors have relied on genetic algorithms, evolutionary strategies or similar approaches that do not require an analytic formulation of the function to be optimized.

A few authors have dealt with the problem of pathology processing pipelines using automatic approaches. Held et al. (2013) successfully optimized pathology processing pipelines using a genetic algorithm approach. They have shown that, even when the number of parameters to be optimized is as low as three, the underlying parameter spaces can show several local performance maxima. Manual optimization strategies in which each parameter is incrementally adjusted may easily become trapped in such local maxima. Their results indicate that the genetic algorithm outperforms other approaches in solving the optimization

problem, albeit at the expense of increased computation time. Teodoro et al. (2016) compared several search optimization techniques, including genetic algorithms, for parameter auto-tuning in tissue image segmentation pipelines.

Angelie et al. (2005) have used genetic algorithms to optimize the quality of organ segmentation in magnetic resonance images; their work shows that the same approach can be used to tailor segmentation pipelines to variations in the pulse sequence used for image acquisition, which is an important practical issue when deploying automated pipelines in clinical settings. Compared to the work by Held et al. (2013), their pipeline has a much larger parameter space, with more than 10 parameters. Both works are focused on sensitivity and/or quality of segmentation, and none of them has taken into account the need to jointly optimize competing quality metrics at the same time.

Anastasio et al. (1998) used genetic algorithms to optimize the parameter of a CAD scheme for the detection of clustered micro-calcifications. Ten parameters important for false positive elimination were identied to be included in the optimization. They designed the cost-function as a combination of the true positive and false positive rate, but did not include segmentation quality. Instead of defining a functional form for the cost function, the authors have directly incorporated preferences about the sensitivity-specicity tradeoff into a discrete grid of numbers that encode the desirability of the set of parameter values that can produce it.

In most recent applications, deep learning approaches, and especially deep convolutional neural networks (CNNs) have emerged as a powerful tool in the medical imaging field (Greenspan et al., 2016; Litjens et al., 2017). With deep learning, the steps of lesion detection, segmentation, feature extraction, and supervised classification can all be realized within the optimization of the same deep architecture, thus alleviating the need to define ad-hoc segmentation algorithm or features. Nonetheless, training a deep CNN still requires the optimization of several hyper-parameters (e.g. number and size of layers, kernel size and stride, learning parameters, and so forth), with respect to a final performance metrics. Therefore, automating hyper-parameter selection will still be a relevant problem in the forthcoming years.

In the medical imaging field, where datasets are relatively small, a common approach is to use a pre-trained architecture and use fine-tuning to adjust the weights for the medical domain (Shin et al., 2016; Tajbakhsh et al., 2016); this reduces, but not completely eliminates, the need for hyper-parameter adjustment, as many are constrained by the pre-trained model of choice. Results from several domains show that this strategy is beneficial especially when limited training data are available (Greenspan et al., 2016).

In the deep learning field, a number of strategies have been proposed to effectively choose hyper-parameters, including random search (Bergstra and Bengio, 2012), evolutionary strategy such as the NEAT and HyperNEAT algorithms (Stanley and Miikkulainen, 2002; Miikkulainen et al., 2017) and Bayesian optimization (Dewancker et al., 2016b,a; Wang et al., 2016). Nonetheless, manual search is still a commonly used technique, because it is easy to implement and exploits researchers' experience to reduce the number of trials, which is useful when training large networks that require extensive computational resources. None of these works deal with the need to define a figure of merit that is appropriate for the task

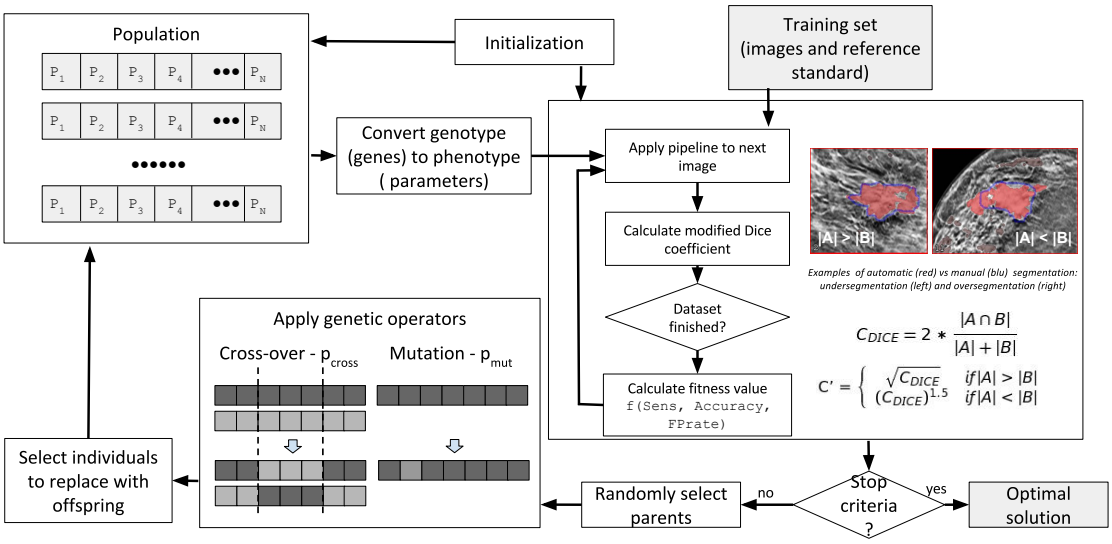

Figure 1: Ovierview of the system architecture. Each individual is encoded as a real-value genome, where each gene represents one parameter in the processing pipeline. At each iteration, the fitness of each individual (or parameter set) is evaluated by computing sensitivity, specificity and segmentation accuracy on a representative dataset. A modified Dice coefficient was used to quantify segmentation accuracy with respect to a reference manual segmentation. The population is evolved using mutation operators such as two-point crossover and mutation.

at hand, and captures all the requirements in the medical imaging field. Litjens et al. (2017) have performed a comprehensive-review of applications of deep learning to medical image analysis; their findings confirm that most researchers employ an intuition-based random search to optimize hyper-parameters.

Finally, a few authors have also sought to optimize acquisition or reconstruction parameters in order to optimize CAD performance (Lau, 2011; Lee et al., 2017). For instance, Lau (2011) developed realistic phantoms to optimize image acquisition parameters and therefore, indirectly, CAD results.In this work, we have assumed the imaging system and reconstruction algorithm to be fixed, which is the most realistic scenario for CAD development both in industry and academia wherever images are acquired by a commercial scanner in a clinical setting. Therefore, our work is a complementary or alternative strategy that can be used in a wide range of applications.

3. Genetic framework for segmentation optimization

This section describes the genetic framework developed in this work, and their implementation in a case study of interest in the medical imaging field. We have implemented the GA in the C ++ language. The software for this work was implemented using the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology. An overview of the proposed architecture is depicted in Figure 1

3.1. Introduction to genetic algorithms

Genetic Algorithms (GAs) are search and optimization algorithms following heuristics process inspired by the mechanisms of natural selection and sexual reproduction (Goldberg and Holland, 1988). GAs work with a set of candidate solutions called a population,where each individual or genome is formed by a string of genes. Both binary or real-value encoding can be used to encode genes.

GAs generate successive populations of alternate solutions, until acceptable results are obtained. A fitness function assesses the quality of each solution in the evaluation step. Individuals with higher scores have higher probability of generating offspring.

The population is evolved by applying genetic operators that introduce random variations in the genetic pool emulating the principle of Darwinian evolution. The main operators that randomly impact the fitness value are crossover and mutation. Crossover, the critical genetic operator that allows new solution regions in the search space to be explored, is a random mechanism for combining the genes of two different individuals; several variations exist such as one point crossover, two point crossover, or homologue crossover. In mutation the genes may occasionally be altered, i.e. in binary code genes changing genes code from 0 to 1 or vice versa. Crossover and mutation probability regulate the frequency with which crossover and mutation are applied.

Offspring replaces the old population, partially or completely depending on the chosen replacement strategy, to form a new population.

3.2. Case study: mass segmentation in digital breast tomosynthesis

As a case study, we selected a simplified version of a Digital Breast Tomosynthesis CAD for the detection and segmentation of masses. A saliency map to highlight masses is calculated on each slice of the Digital Breast Tomosythesis after downsampling by a factor of four, and applied Contrast Limited Adaptive Histogram Equalization (CLAHE). Methods for detecting blob patterns in a multiscale setting using Laplacian of Gaussian (LoG) have been proposed to detect lesions that exhibit blob-like structures such as pulmonary nodules (Diciotti et al., 2008).

Following the principles of the linear scale-space theory, the contribution of the blob detector at each scale s can be defined as:

$$L(x,y,s) = \begin{cases} I(x,y) \bigotimes LoG_s(x,y) & if I(x,y) \bigotimes LoG_s(x,y) > 0 \\ 0 & if I(x,y) \bigotimes LoG_s(x,y) < 0 \end{cases} \quad (1)$$

where $I(x,y)$ is the image at position (x,y) and $LoG_s(x,y)$ is the Laplacian of Gaussian kernel at scale s. The scale parameter s is defined as the width of the central lobe $s = 2\sqrt{(2)}\sigma$, where σ is the standard deviation of the Gaussian kernel. With proper normalization, the volume of the central lobe of the LoG kernel is independent of σ and has a positive sign. In this way, it is possible to compare responses with different values of σ, or in other words, at different scales.

The saliency map $S(x, y)$ is obtained by integrating the scale contribution space $L(x, y, s)$ between a minimum scale m and a maximum scale M:

$$S(x, y) = \int_m^M L(x, y, s)ds \tag{2}$$

In practice, the calculation of the integral is discretized and a finite number N of scales are considered. The discretization step is defined as

$$\Delta s = \frac{M - m}{N} \tag{3}$$

A region growing algorithm is applied to the saliency map to detect lesion candidates and perform an initial segmentation. The region growing algorithm depends on two parameters: the seed threshold T_s and the growing threshold T_g . Both thresholds as defined as the $n - th$ percentile of the image to account for variations between images and assuming that, within a given image, lesions are always among the the objects with the highest activations in the saliency map. Pixels with intensity higher than T_s (or better, higher than the intensity corresponding to the T_s percentile) will form the objects seeds; from the seeds, segmentation will be expanded to adjacent pixels with intensity higher than T_g . This will not achieve perfect segmentation - hence a further refinement step will be needed. However, for the purpose of this paper, we will focus only on the choice of the parameters m, M, Δs , T_s , T_g .

3.3. Parameter encoding and constraints

In our case, a genome is constituted by five real-valued genes, each encoding one of the parameters of the processing pipeline. Real-value encoding, while a natural choice in this case, yields very large search space, which can be restricted by applying prior knowledge about the range of possible values for each gene. Lower and upper bounds can be determined based on theoretical, as well as practical constraints (e.g. reducing computation time). Examples of lower and upper bounds for this case study are shown in Table 1. Upper and lower bounds for m and M were established based on the distribution size of the target lesions. Bounds for the seed and growing thresholds were determined based on results from previous manual optimization, as well as qualitative examination of saliency maps. All parameters were discretized to further reduce the search space, as small differences in parameter values do not yield appreciably different results.

Parameter values are often inter-dependent and certain relationships may need to be satisfied for a solution to be viable. In our case study, order relationships must be verified: $m < M$ and $T_s > T_g$. Furthermore, the time needed to calculate the saliency map is linearly dependent on the number of scales, that is given by

$$N = \frac{M - m}{\Delta s} \tag{4}$$

Hence, we assumed that $1 < N < 25$ to constraint the overall calculation time. Genomes that do not satisfy these constraints are not viable solutions to the problem; a common way to implement constrained optimization in Genetic Algorithms is by penalizing unviable solutions, as explained in Section 3.4.

Parameter	Lower Bound	Upper Bound	Discretization step
Experiment 1			
m	3	10	0.5
M	7	30	0.5
Δs	0.5	3	0.5
T_s	0.92	0.99	0.001
T_g	0.85	0.97	0.001
Experiment 2			
m	3	30	0.5
M	3	30	0.5
Δs	0.5	5	0.5
T_s	0.7	0.99	0.001
T_g	0.7	0.97	0.001

Table 1: Ranges and discretization steps for each parameter. In Experiment 1, we selected a narrow search space based on available domain knowledge; in experiment 2, we selected a larger parameter space to analyze the repeatably of the experiment

3.4. Fitness function

The performance of a detection and segmentation algorithm depends on its sensitivity, specificity (i.e. false positive rate) and quality of the final segmentation. The fitness function was thus defined as the weighted sum of three terms:

$$f = w_1 * Sens + w_2 * Acc + w_3 * FP_{rate} \tag{5}$$

where $Sens$ is sensitivity, Acc is the average lesion segmentation accuracy, and FP_{rate} is related to the average false positive rate. The weights w_1,w_2 and w_3 are used to determine the relative importance of each figure of merit in the overall fitness. In our case, we chose values of 0.5, 0.35 and 0.15, respectively, placing much more emphasis on accurate and reliable lesion segmentation than low false positive rate, because we assumed that a subsequent false positive classification steps would be use to increase specificity.

Both sensitivity and segmentation accuracy are defined with respect to an annotated reference standard. It is assumed that, for each lesion, a manual segmentation is available to be able to evaluating quality of segmentation. One of the most common figure of merits for segmentation accuracy is the Dice coefficient (Zou et al., 2004).

First of all, the set of findings identified by the algorithm are compared to the reference standard and labelled as false or true positive findings; since a manual segmentation is assumed to be available, a strict overlapping criterion is employed to discriminate true from false positive findings (i.e. findings overlapping at least one manual segmentation are considered true positive findings). Each finding is then compared against its associated lesion to evaluate the quality of segmentation.

If A is the set of voxels in the reference standard, and B is the set of voxels in the corresponding finding segmented by the pipeline, the Dice coefficient measures the ratio

of the overlap between the reference standard and the segmentation, with respect to their union:

$$C_{DICE} = 2 * \frac{|A \cap B|}{|A| + |B|} \tag{6}$$

Values close to 1 indicate well segmented lesions, values close to 0 indicate poorly segmented lesions and values equal to 0 are associated to false negatives.

One of the problem of the Dice coefficient is that under- and over-segmentation are treated equally. In our case study, we believe that over-segmentation is often more troublesome than under-segmentation, as it may affect feature extraction from segmented lesions, and may be particularly confusing for the radiologist. If the segmented object is much larger in volume than the reference segmentation, it is counted as a false positive, rather than a true positive, by assigning a value of 0 to the modified Dice index; this penalizes any matching that may occur purely by chance (i.e. the segmentation finds so many false positives, that a number of lesions are detected by chance alone). In our specific case study, a final segmentation refinement step is performed, that may further expand under-segmented lesions if needed. To take into account these preferences, we have modified the Dice coefficient as follows:

$$C' = \begin{cases} \sqrt{C_{DICE}} & if |A| > |B| \\ (C_{DICE})^{1.5} & if 0.4 \leq \frac{|A|}{|B|} < 1 \\ 0 & if \frac{|A|}{|B|} < 0.4 \end{cases} \tag{7}$$

Since the value of the Dice coefficient is comprised between 0 and 1, the modified Dice coefficient is increased when the segmented object is smaller than the reference segmentation, and decreased when the segmented object is larger than the reference segmentation.

The individual false positive rate FP^j_{rate} for each volume j is determined by calculating the fraction of the total volume of false positive findings over the total breast volume occupied by false positive findings. Let N_L be the number of lesions in the training dataset, and N_V the number of image volumes. Then the contributions to the fitness function are defined as follows:

- Sensitivity $Sens$ is defined as the number of detected lesions over the total number of lesions N_L. To increase accuracy and improve convergence only lesions with modified Dice coefficient $C' > 0$ are counted as detected;
- Segmentation accuracy Acc is defined as the average modified Dice coefficient C' over all lesions;
- Average false positive rate FP_{rate} is defined as $1 - \frac{\sum_{j=1}^{N_V} FP^j_{rate}}{N_V}$.

In order to better outline the difference between the Dice coefficient and our modified implementation, we have calculated the theoretical values of both indices for two circles of ray r_1 and r_2, respectively. As shown in Figure 2, the two values diverge rapidly in the case of oversegmentation. The exponents were chosen experimentally by comparing the modified Dice scores with a qualitative visual assessment of segmentation quality.

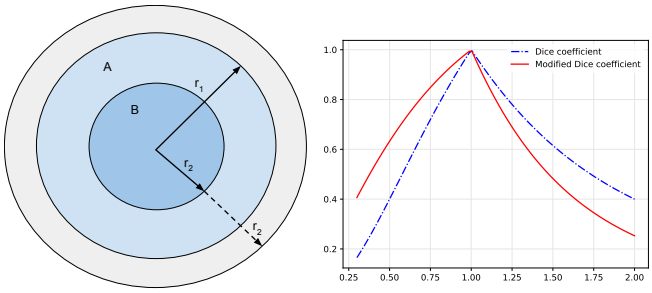

Figure 2: Right: Values of the Dice and modified Dice coefficients for two circles of ray r_1 and r_2, respectively, where $r_1 = 1$ and r_2 varies between 0.2 (undersegmentation) and 2.0 (oversegmentation). Left: graphical representation of the two circles A and B, of ray r_1 and r_2 respectively, where A represents the reference segmentation and B the output of the segmentation algorithm.

3.5. Unfeasible solutions

Our problem lays in the class of constrained optimization problems, as defined in Section 3.3. While cross-over and mutation can be modified to constraint parameter values within the ranges defined in Table 1, it cannot be guaranteed that all generated offspring will lay in the feasible space.

The most common approach to deal with constrained optimization in genetic algorithms is to define a penalty function that lowers the fitness value of the unfeasible solutions. The simplest penalty mechanism is rejection by death penalty: unfeasible individuals are penalized by using a constant, very low fitness function (0 in our case) (Dasgupta and Michalewicz, 2013; Delsanto et al., 2004, 2006).

Rejection by death penalty was a very natural choice in our case, because in some cases (e.g. when $M < m$) it is merely not possible to calculate the fitness function. It is also very computationally efficient as it reduces the number of solutions to evaluate and the genetic algorithms quickly converges to a range of feasible solutions. This simple heuristics has been shown to work acceptably well when the feasible regions are convex and relatively large, which is a reasonable assumption in our case (Dasgupta and Michalewicz, 2013).

In other cases, our approach would have limitations. For example, in search problems where there are small, not convex feasible regions, it might be essential to improve unfeasible solutions, as opposed to rejecting them. Moreover, quite often the system can reach the optimum solution easier if it is possible to cross an unfeasible region (especially in non-convex feasible search spaces) (Dasgupta and Michalewicz, 2013; Montemurro et al., 2013).

4. Material and methods

4.1. Dataset

The dataset used in these experiments included 132 cases (corresponding to 437 volume reconstruction) with histology-proven, screen-detected malignant lesions. Mammographic examinations included in this study were randomly selected from a previous institutional review board-approved prospective trial, and all subjects gave informed consent.

All cases included craniocaudal (CC) and mediolateral oblique (MLO) views acquired on a Selenia Dimensions (Hologic Inc, Bedford, MA, USA). The dataset includes 68 malignant soft tissue lesions (61 masses and 7 architectural distortions) proven either by final histology or micro-histology. Average lesion size ($\pm$ standard deviation) was 22 $\pm$ 11 mm (range 6 - 60 mm); in most cases (54), both MLO and CC views were available and showed the lesion; in the remaining 14 cases, only the MLO projection was acquired (7 cases) or the lesion was visible on one view only (7 cases).

An experienced breast radiologist annotated lesions by drawing a three-dimensional bounding box using the im3D CAD BREAST DTS reading workstation (im3D, Turin, Italy), based on all information available for the case, including mammography, ultrasound (available in all cases) and biopsy reports. The radiologist was instructed to draw the bounding box as close as possible to the lesion, including spiculations; a two-dimensional bounding box was initially drawn on the central slice and then extended to all the slices in which the lesion was visible and in focus.

For the purpose of segmentation optimization, using the entire dataset was too computationally expensive. In order to evaluate the objective function for each individual, we have to run the pipeline and evaluate the results for each case in the training dataset. The computation time of the saliency map varies according to the number of scales used: in our experimental setup, computation times varied between 45 seconds and 4.5 minutes, corresponding respectively to 2 and 24 scales.

Therefore, we randomly selected a training dataset containing a smaller subset of 16 lesions: lesions were stratified in order to include an equal number of small ($< 15mm$) and large ($> 15mm$) lesions; this artificially balances the training set in order to ensure that we do not overfit parameters to very large or very small lesions. For the training set, a researcher manually segmented the lesion based on radiologists reports, as the bounding box was not sufficient to assess segmentation quality.

The final sensitivity was assessed on the entire dataset, that was used as independent testing.

4.2. Experimental setup

An initial population of 100 randomly initialized individuals is created and allowed to evolve for 20 generations using a steady state technique using the training dataset. Individuals which can potentially mate are selected according to their fitness score using a standard roulette-wheel selection scheme. Crossover is performed with probability $p_{cross} = 0.8$, using two-points crossover. Each new individual may mutate with probability $p_{mut} = 0.1$, that is each gene has a probability p_{mut} of being replaced by a random value. At each generation, 80% of the entire population is replaced. There is not a universally accepted criteria for determining convergence of a genetic algorithm; we defined convergence as the ratio of the 5th previous best-of-generation score to the current best-of-generation score being greater than 0.99, and standard deviation of the population less than 0.001.

Assuming an average time of calculation of 2.5 minutes and a dataset of 16 cases, the genetic algorithm employs up to 40 minutes to evaluate the fitness of a single individual and up to 4000 minutes (or about 66 hours) for the entire population. In practice, this is an

upper bound since genomes that do not fulfill parameter constraints (see Section 3.3) will be discarded without having to compute the saliency map.

In order to reduce the search space and increase the probability of convergence, each parameter was bound and discretized. In an initial experiment (see Table 1, Experiment 1) we chose smaller search spaces that were derived from available domain knowledge. The upper and lower bounds for m and M were established based on the distribution size of the target lesions (smallest and largest scale included in the saliency map integration); masses smaller than < 3 mm are seldom detectable, while very large lesions (>3 cm) are generally easy to spot for the radiologist. The ranges of the seed and growing thresholds were chosen based on our previous experience derived from an initial manual exploration of the parameter space. However, given that the seed and growing thresholds are defined in terms of the n-th percentile of the saliency map distribution, if their values are too low a large portion of the image would be segmented with very little utility for the radiologist. For instance, $T_s = 0.92$ implies that only pixels with saliency values in the top 8% are selected as seeds. Furthermore, since both m and M, and T_s and T_g need to satisfy order constraints (see Section 3.3), very large values of m and T_g are unlikely. For all parameters, the discretization step was chosen based on an initial analysis of how steep is the change in the final segmentation with respect to a change in the algorithm parameter.

In other more complex algorithms, it might not be possible or advantageous to restrict the search space in this way. Therefore, we repeated the experiment relaxing some of our initial assumptions and including a much larger search space. The main purpose of this analysis was to ensure that the GA could still converge to a solution as least as good as the previous one.

4.3. Data analysis

The best individual selected by the genetic algorithm was tested on the entire data set. Per-lesion sensitivity and associated 95% confidence intervals (CIs) were calculated after the CAD candidates and radiologists bounding boxes were automatically matched. Matching criteria were as follows: a mass or architectural distortion was detected if the radiologist bounding box overlapped with a mass CAD bounding box (or a combination of) by at least 6% in volume, and 20% along the direction perpendicular to the detector. A lesion was considered detected if identified by CAD on either the CC or MLO view, or in both.

We also collected and analyzed all distinct viable solutions detected by the algorithm. In order to assess the relationship and influence between each parameter and the three components of the fitness function, an exploratory analysis was conducted using both correlation analysis and association rules. Association rule learning is a machine learning method for discovering interesting relations or set of frequent pattern between variables in large databases (Agrawal and Srikant, 1994). Association rules do not require any assumption on the statistical distribution of each parameter / variable, nor on their conditional distribution. Association rules were first defined for basket analysis, to determine how often pairs or combination of items appeared together in a market transaction, but can be applied to a wide range of problems (e.g., document summarization (Baralis et al., 2016), network traffic analysis (Apiletti et al., 2013)). In our case, each "transaction" represents a genome, that

is a potential solution, along with its figures of merit and fitness function. Non-binary variables are converted to a series of binary items through binning, so that each item represents whether a given gene (i.e. parameters) falls within a predefined range. An association rule is an implication expression of the form $X \rightarrow Y$, where X (the antedecent) and Y (the consequent) are disjoint set of items. For instance, one of the association rule extracted in our problem was $M = [7 - 12.750] \rightarrow Acc = [0.601 - 0.75]$, meaning that a value of M between 7 and 12.750 is associated to an accuracy between 0.6 and 0.75. Several transactions (i.e. individual) may contain an itemset; in other words, several solutions will be characterized by those values of M and accuracy. The relevance and strength of each association rule can be measured through its support, confidence, and lift. Support $s(X \rightarrow Y)$ measures how often a given association appears in a dataset: rules with a high support are less likely to appear by chance only. Confidence $c(X \rightarrow Y)$ measures how frequently items in Y appear in individuals that contain X: it measures the strenght of the inference and provides an estimate of the conditional probability of Y given X. The lift index was introduce to measure the relevance or interestingness of an association rules, and measures the (symmetric) correlation between antecedent and consequent of the extracted rules (Agrawal and Srikant, 1994). It is defined as the ratio between the rule's confidence, and the support of the rule consequent:

$$Lift(X \rightarrow Y) = \frac{c(X \rightarrow Y)}{s(Y)} \tag{8}$$

Lift provides a measure of the true strength of an association by correcting for the expected confidence level given the prevalence of X and Y in the dataset, similarly to Cohen's Kappa coefficient of diagnostic agreement. A lift value greater than (lower than) 1 indicates that the antecedent and consequent appear more (less) frequently together than expected, thus indicative a positive or negative correlation. The interest of rules having a lift value close to 1 may be marginal, as they are the most likely to occur by chance.

To extract association rules, each attribute was discretized using automatic or, when more appropriate, user-supplied binning. The Frequent Pattern-growth algorithm implemented in RapidMiner was used to obtain frequent itemset (RapidMiner, Accessed: December 2015); then association rules are extracted and those with support greater than 0.15 and confidence greater than 0.7 are reported. Extracted rules were revised by a researcher with experience in the field of CAD design (L.M.) and pruned to limit the length of the antecedent (two parameters) and the consequent (at most one parameter), as very long rules were more difficult to interpret.

5. Results

5.1. Optimization results

The genetic algorithms was allowed to evolve for 20 generations, and reached convergence around generation 17. The evolution of fitness values across the entire population is reported in Figure 3.

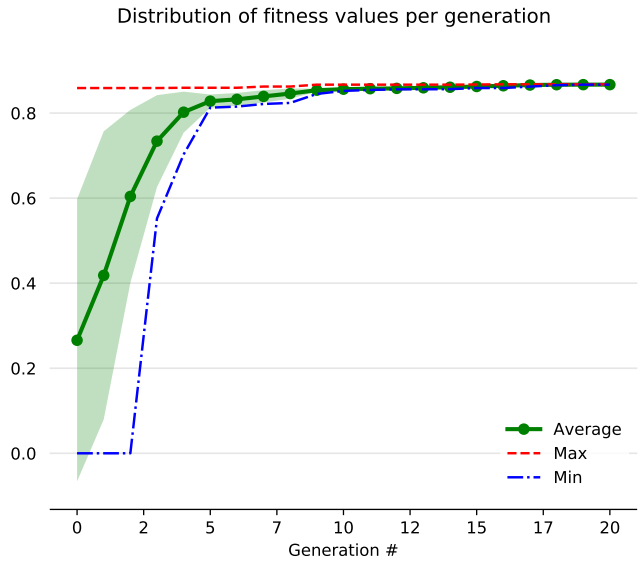

Figure 3: Minimum, maximum (dashed lines) and average (solid line) fitness values for each generation. Average fitness ± standard deviation are represented by the solid line. The first generation is excluded from the plot for reasons of clarity: within the first two generations, unfeasible solutions are discarded quickly and hence the fitness range is much larger. Note that convergence is achieved at generation 17.

The best parameter configuration was as follows: m = 4, M = 15, $\Delta s = 1.5$, $T_s = 0.951$ and $T_g = 0.928$. Previous manual optimization of the parameters yielded values m = 3, M = 20, $\Delta s = 1$, $T_s = 0.946$ and $T_g = 0.93$. On the smaller dataset used for parameter optimization, which included 16 lesions, this optimal solution corresponds to a 100% sensitivity, a 0.63 average modified Dice coefficient (lesion accuracy) and a $FP_{rate} = 0.986$ (i.e. on average 2.4% of the breast volume was segmented as false positive). The segmentation with highest fitness value is also optimal with respect to sensitivity and segmentation accuracy; the false positive rate was not optimal but better than average, with a value close to the 65th percentile.

On the entire dataset, which included 68 lesions, of which 50 were not used for training, the optimal solution yielded a 100% sensitivity based on the matching criteria described in Sections 4.3 and 4.1.

5.2. Exploration of the parameter space

Overall, the GA found 382 distinct viable solutions; additionally, 144 solutions were discarded based on parameter constraints. The distribution of each parameter in the viable solution space is represented in Figures 4.

In Figure 5, the distribution of the genome fitness, along with that of the sensitivity, average segmentation accuracy and average FP_{rate} of each individual is reported. A low number of poor solutions exhibit unacceptable sensitivity and hence their sensitivity score was set to 0. Overall, most solutions explored by the GA had sensitivity greater than 80%, and roughly 20% of all viable solutions had 100% sensitivity.

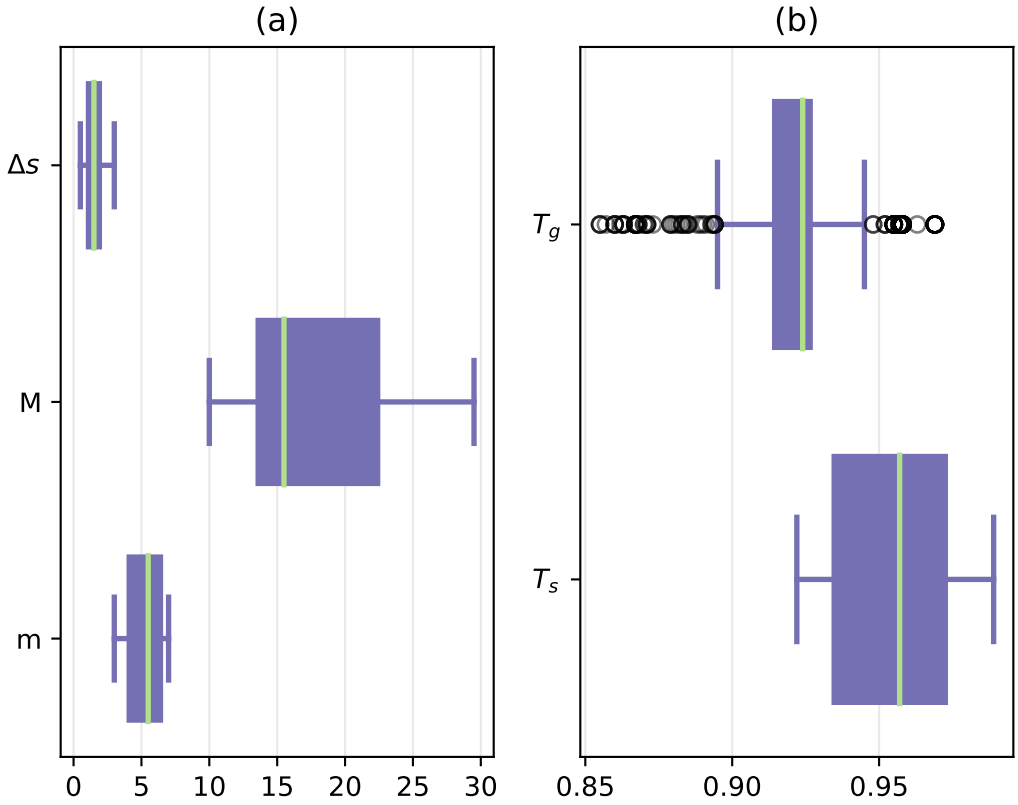

Figure 4: Box plots of the parameter values: saliency map parameters (a) and region growing parameter (b). The range of each parameter is very close to the range selected for each gene. Optimal values found by the GA were m = 4, M = 15, $\Delta s = 1.5$, $T_s = 0.951$ and $T_g = 0.928$.

Correlation analysis was performed to evaluate the strength of the relationship between each parameter and each component of the fitness function, as shown by the heatmap in Figure 6. The highest correlation coefficients were found for parameters M and T_g . M is negatively correlated with overall fitness score (-0.27), as well as sensitivity (-0.18) and accuracy (-0.22). T_g is positively correlated with all outcome measures, with values 0.60, 0.41, 0.65 and 0.65, respectively. Notably, the individual correlation of M and Δs with outcome measures are negative. Segmentation accuracy Acc was positively correlated with sensitivity (0.99) and the inverse of the FP_{rate} (0.85), which means that higher segmentation accuracy will yield a moderately larger number of falsely segmented pixels. This high correlation is consistent with the fact that the Dice coefficient was modified to "punish" oversegmentation, and the false positive rate is volumetric, hence false positives with larger volumes are also "punished" with a lower value. Both the volume of false and true positives is controlled by the growing threshold T_g .

5.3. Association rules

Association rules were extracted using the approach described in Section 4.3; rules with support greater than 0.15 and confidence greater than 0.65 are reported in Table 2.

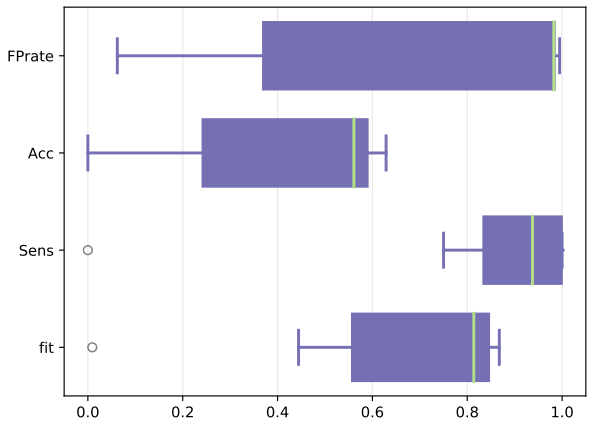

Figure 5: Distribution of the genome fitness values and of each of their components: sensitivity *sens*, average lesion segmentation accuracy *Acc* and average false positive rate FP_{rate}

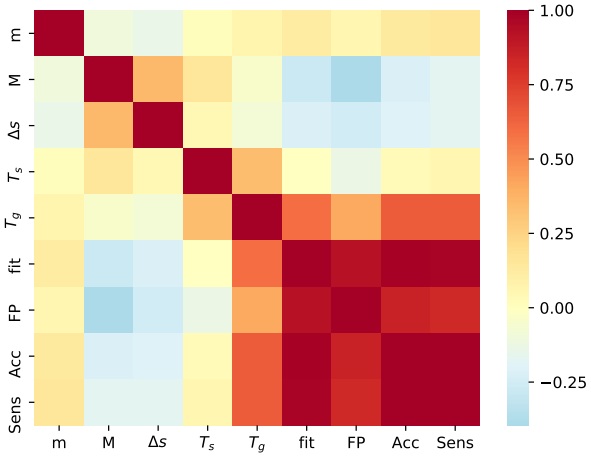

Figure 6: Correlation coefficients between each parameter and quality of the pipeline outcome: overall fitness value (f), sensitivity (sens), average lesion segmentation accuracy (Acc) and average false positive rate FP_{rate}

No.	Antecedent	Consequent	Support	Confidence	Lift
1	$\Delta s = [0 - 1.75]$, $Sens = 100\%$	$M = [12.75 - 18.5]$	0.188	0.735	1.559
2	$T_g = [0.908 - 0.927]$, $Sens = 100\%$	$M = [12.75 - 18.5]$	0.173	0.742	1.574
3	$\Delta s = [0 - 1.75]$, $Acc = [0.501 - 0.60]$	$M = [12.75 - 18.5]$	0.280	0.743	1.273
4	$M = [12.75 - 18.5]$, $Acc = [0.501 - 0.60]$	$T_g = [0.908 - 0.927]$	0.238	0.784	1.320
5	$\Delta s = [0 - 1.75]$, $M = [12.75 - 18.5]$	$T_g = [0.908 - 0.927]$	0.330	0.797	1.342
6	$Acc = [0.501 - 0.60]$, $Sens = 0.9375$	$\Delta s = [0 - 1.75]$	0.217	0.798	1.077
7	$T_g = [0.908 - 0.927]$, $m = [3 - 5.333]$	$\Delta s = [0 - 1.75]$	0.183	0.814	1.099
8	$T_g = [0.908 - 0.927]$, $Acc = [0.501 - 0.60]$	$M = [12.75 - 18.5]$	0.238	0.827	1.756
9	$M = [12.75 - 18.5]$, $m = [3 - 5.333]$	$\Delta s = [0 - 1.75]$	0.175	0.838	1.130
10	$\Delta s = [0 - 1.75]$, $T_s = [0.92 - 0.937]$	$T_g = [0.908 - 0.927]$	0.212	0.844	1.420
11	$M = [12.75 - 18.5]$, $Sens = 100\%$	$T_g = [0.908 - 0.927]$	0.173	0.880	1.481
12	$M = [12.75 - 18.5]$, $Acc = [0.501 - 0.60]$	$\Delta s = [0 - 1.75]$	0.280	0.892	1.204
13	$T_g = [0.908 - 0.927]$, $T_s = [0.92 - 0.937]$	$\Delta s = [0 - 1.75]$	0.212	0.900	1.215
14	$T_g = [0.908 - 0.927]$, $Acc = [0.501 - 0.60]$	$\Delta s = [0 - 1.75]$	0.267	0.927	1.252
15	$\Delta s = [0 - 1.75]$, $Sens = 0.9375$	$Acc = [0.501 - 0.60]$	0.217	0.933	2.096
16	$M = [12.75 - 18.5]$, $Sens = 0.9375$	$Acc = [0.501 - 0.60]$	0.188	0.935	2.101
17	$M = [12.75 - 18.5]$, $T_s = [0.92 - 0.937]$	$\Delta s = [0 - 1.75]$	0.160	0.938	1.267
18	$T_g = [0.908 - 0.927]$, $M = [12.75 - 18.5]$	$\Delta s = [0 - 1.75]$	0.330	0.940	1.269
19	$M = [12.75 - 18.5]$, $T_s = [0.92 - 0.937]$	$T_g = [0.908 - 0.927]$	0.162	0.954	1.605
20	$M = [12.75 - 18.5]$, $Sens = 100\%$	$\Delta s = [0 - 1.75]$	0.188	0.960	1.296
21	$T_g = [0.908 - 0.927]$, $Sens = 100\%$	$\Delta s = [0 - 1.75]$	0.225	0.966	1.304

Table 2: Association rules extracted using the Frequent Pattern-growth algorithm. Only rules with support greater than 0.15 and confidence greater than 0.7 are reported.

Rules that have highest lift, support and confidence are No. 5, 8, 12, 15, 16, 18, 19, 20, and 21,as shown in Table 2; rules 8, 15, 16, and 19 are associated with the highest lift values. In many rules, high sensitivity is associated with high segmentation accuracy (e.g 15 and 16); this is not surprising given the nature of the algorithm, as well as the fact that low quality segmentation were excluded from analysis. It has to be pointed out that evaluating segmentation accuracy is much more expensive than sensitivity, for which establishing ground truth is less time consuming. Rules 14, 15, 16 all indicate that segmentation accuracy between 0.501 and 0.6 is associated with specific value ranges of the pipeline parameters. This range corresponds to a sufficient to good segmentation accuracy, and approximately half of the solutions investigated by the GA fall within this range. In all rules, a value of M between 12.75 and 18.5 is associated with this level of segmentation accuracy; this is consistent with correlation analysis as M is negatively correlated with all outcomes. In even simple pipelines complex dependencies between parameters may arise. This is exemplified by rules 5, 8, 9, 13, 17, 18 and 19: for instance, specific values of M and T_g (or M and Δs) are most likely associated with specific values of Δs (or T_g), thus indicating that the GA correctly tailored specific threshold values to the saliency map parameters. Notably, this may also indicate that the genetic algorithm has explored most frequently parts of the parameter space that are associated with good segmentation values, which is consistent with the high convergence speed of the genetic algorithm. Overall, the rules indicate that when the values of M, m, T_s and Δs lie in the intervals $[12.85 - -18.5]$, $[3 - 5.333]$, $[0.92 - 0.937]$ and $[0 - 1.75]$, then segmentation accuracy will be generally higher. Remarkably, the best solution found was not found exactly within this range, suggesting the existence of multiple regions of good solutions.

5.4. Stability with respect to fitness function parameters

Individual ranking depends on the relative weights w_1,w_2 and w_3 that are used to combine several figure of merit in the overall fitness (see Equation 5), which were set to values of 0.5, 0.35 and 0.15, respectively.

Since running an experiment for each possible triplets of values was unfeasible, we performed a sensitivity analysis on all viable solutions found by the GA. Specifically, we sought to determine whether their relative ranking, and specifically the optimal solution, changed as a function of w_1,w_2 and w_3. This strategy, while not conclusive, is indicative of what would happen for at least the portion of the search space explored by the GA. Values for w_1,w_2 and w_3 were allowed to vary between 0.01 and 0.99.

A limited number of possible solutions was found:

- m = 4, M = 15, $\Delta s = 1.5$, $T_s = 0.951$ and $T_g = 0.928$
- m = 3, M = 16, $\Delta s = 2$, $T_s = 0.961$ and $T_g = 0.933$
- m = 6.5, M = 23.5, $\Delta s = 2$, $T_s = 0.959$ and $T_g = 0.957$
- m = 6.5, M = 22.5, $\Delta s = 3$, $T_s = 0.984$ and $T_g = 0.969$
- m = 4, M = 22.5, $\Delta s = 1$, $T_s = 0.984$ and $T_g = 0.969$

- m = 6.5, M = 15.5, $\Delta s = 1$, $T_s = 0.984$ and $T_g = 0.969$
- m = 7, M = 22.5, $\Delta s = 2.5$, $T_s = 0.871$ and $T_g = 0.952$
- m = 6.5, M = 11.5, $\Delta s = 2$, $T_s = 0.959$ and $T_g = 0.957$
- m = 6.5, M = 14, $\Delta s = 0.5$, $T_s = 0.928$ and $T_g = 0.927$

On the training set, all solutions have sensitivity greater than 90%, segmentation accuracy between 0.56 and 0.63, and FPrate between 0.983 and 0.994.

5.5. Stability with respect the search range

In this experiment, we re-run the genetic algorithm search on a larger parameter space, as defined in Table 1. The best parameter configuration was as follows: m = 5, M = 16.5, $\Delta s = 2$, $T_s = 0.943$ and $T_g = 0.937$; this solution has a fitness value of 0.868, corresponding to 100% sensitivity, 0.63 accuracy and $FP_{rate} = 0.986$. While the solution is slightly different to the best solution found in experiment 1 (m = 4, M = 15, $\Delta s = 1.5$, $T_s = 0.951$ and $T_g = 0.928$), the two solutions are equivalent in terms of fitness, as they produce almost equivalent segmentation.

The evolution of fitness values across the entire population is reported in Figure 7. We also reported the distribution (boxplots) of the parameters and scores of all 303 viable solutions found by the GA (see Figures 8 and 9). With a larger search space, the population diversity is larger in initial generators, but the genetic algorithms converges nonetheless to the same region of the parameter space in a comparable number of generations (17). A larger number of non-viable solutions is explored.

6. Discussion

CAD systems, like many other image processing tasks, include a detection and segmentation processing pipeline whose performance is dependent on the value of many parameters. Successful optimization is critical to achieve high quality performance. In this paper, we designed an optimization framework for detection and segmentation pipelines, based on genetic algorithms. As case study, a simplified CAD system for mass detection in Digital Breast Tomosynthesis (DBT) was selected.

Defining a suitable figure of merit (e.g. fitness function) is crucial to successful optimization. In a CAD system, there are two fundamental ways to assess quality of segmentation: directly, by computing/measuring spatial overlap with a known ground truth, or indirectly, by computing the performance of the entire pipeline including the classification or false positive reduction scheme. In the latter case, the ROC or FROC curve (which plots the fraction of correctly localized lesions as a function of the average number of false positives per image) are generally used to assess the overall quality of a CAD system (Petrick et al., 2013).

In this paper, we focused on the detection and segmentation step, by defining a fitness function that includes three terms: sensitivity, false positive rate and average segmentation quality. Segmentation quality was computed by measuring spatial overlap with a manual segmentation (ground truth). We defined a modified Dice coefficient that penalizes

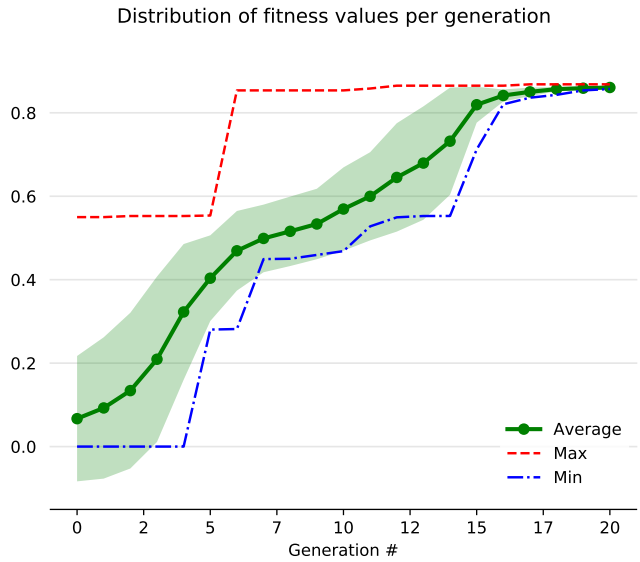

Figure 7: Minimum, maximum (dashed lines) and average (solid line) fitness values for each generation. Average fitness ± standard deviation are represented by the solid line. The first generation is excluded from the plot for reasons of clarity: within the first two generations, unfeasible solutions are discarded quickly and hence the fitness range is much larger. Note that convergence is achieved at generation 17.

over-segmentation compared to under-segmentation. In our experience, the modified Dice coefficient is more consistent with perceived quality of the resulting segmentation. Moreover, a large mark that covers a large portion of the breast has a higher chance to overlap with a reference lesion, but the CAD system may actually have missed the true lesion or may have segmented large parts of the breast parenchyma as well. Marks labeled as true positives that are unlikely to attract the attention of the reader to the abnormality are worrisome as they may lead to inflated standalone CAD performance estimates (Petrick et al., 2013).

Usually, a reference standard is established through manual annotation of a training and testing set, in which each lesion is identified by a simple mark, either the centroid or a bounding box (Petrick et al., 2013). Providing an accurate manual segmentation is much more time consuming, and also operator dependent. Is it thus necessary or useful to assess the quality of the segmentation in CAD applications?

In general, the answer is probably yes if the CAD output is directly used for quantitative measurements, such as lung nodule volumetry (van Ginneken et al., 2011). Radiologists may trust CAD more if they perceive that lesions are accurately segmented.

On the other hand, some authors have argued that the performance metrics should be more closely tied to the specific clinical task. In practice, in the vast majority of CAD systems the last step is false positive reduction based on feature extraction and a statistical classifier. If feature extraction is affected deeply by the quality of the segmentation, the optimal segmentation may be the one that maximizes the final classifier performance, rather than overlap with a reference manual segmentation. Work by Kuo et al. (2014) suggests that subtle changes in segmentation quality may produce larger differences in classification

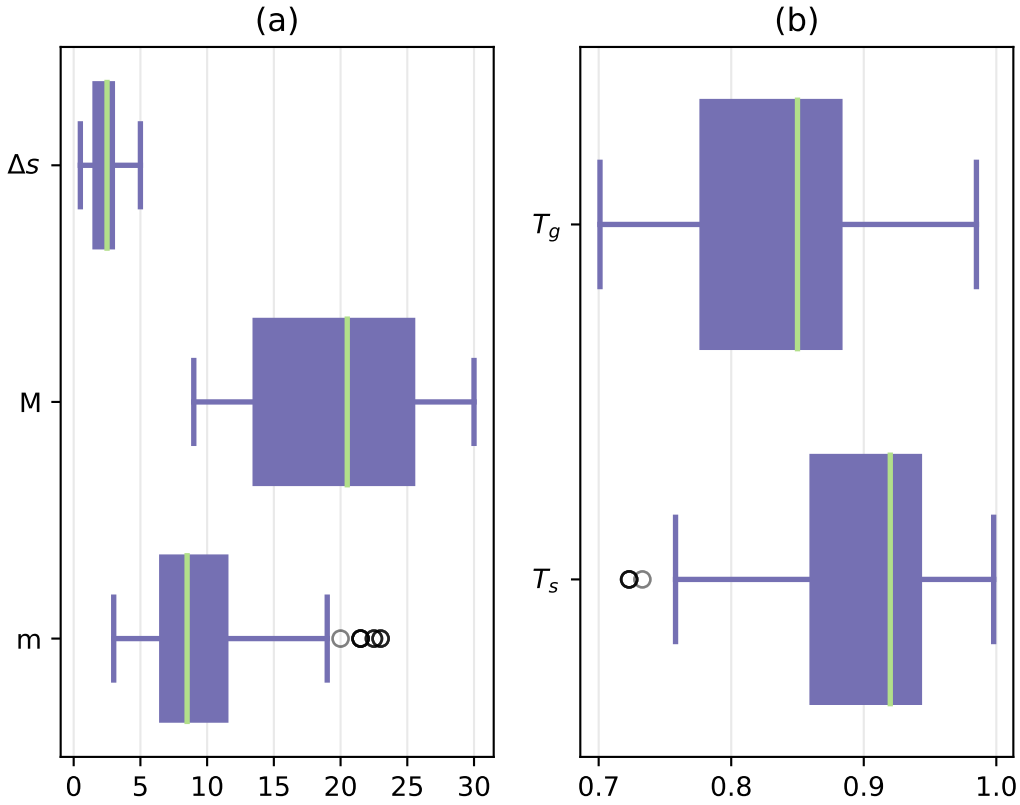

Figure 8: Box plots of the parameter values: saliency map parameters (a) and region growing pararameter (b). The range of each parameter is very close to the range selected for each gene, except for M where few solutions have values 25. Notably, the optimal values are very close to the average value for all parameters except Δs. Optimal values found by the GA for this experiment were m = 5, M = 16.5, $\Delta s = 2$, $T_s = 0.943$ and $T_g = 0.937$

accuracy (as measured by the area under the ROC curve) than in spatial overlap indices.

By modifying the fitness function, the proposed framework can be easily adapted based on the available ground truth and the desired performance metrics. If manual segmentation were not available, sensitivity and specificity can still be calculated by selecting an appropriate mark-labeling rule, as discussed by Petrick et al. (2013). The mark-labeling rule should be selected in a such a way that sizes of the true positive marks are comparable to the lesion sizes as determined by the reference standard (Petrick et al., 2013). In our simple case study, higher sensitivity was generally associated with higher segmentation accuracy, but this may not hold true for more complex pipelines. Furthermore, segmentation accuracy may account for subtle variations that are not captured by sensitivity. Otherwise, the area under the ROC curve can be used as an alternative figure of merit, especially if feature calculation/selection and classifier training are included in the pipeline (Kuo et al., 2014).

Notably, genetic algorithms have been used by several authors to choose optimal feature subset (feature selection) and to optimize meta-parameters for classifier training (Huang and Wang, 2006; Agliozzo et al., 2012).

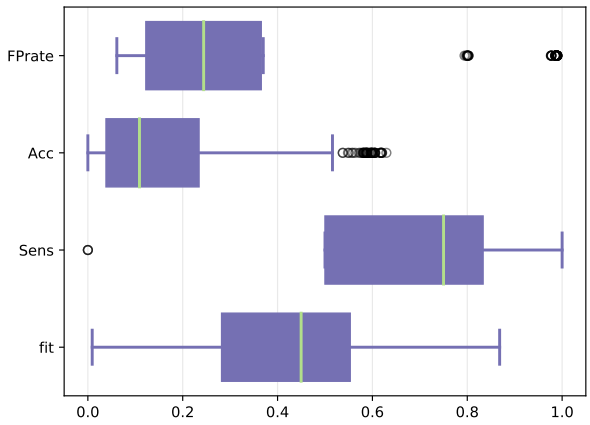

Figure 9: Distribution of the genome fitness values and of each of their components: sensitivity $sens$, average lesion segmentation accuracy Acc and average false positive rate FP_{rate}

In complex image processing pipeline, manual exploration of the parameter space has been shown in literature to yield sub-optimal results even with low number of parameters (Angelie et al., 2005; Held et al., 2013). Usually, manual optimization is performed by adjusting one parameter at the time, which is a greedy optimization technique that does not guarantee finding a global minimum. In our case study, the genetic algorithm converged to a good solution in a low number of iterations. Automating algorithm tuning is an important step to design general purpose frameworks that can be easily adapted to specific tasks, as it allows a reproducible, reliable and less expensive way to assess the effectiveness of a given algorithm on an image database or adapt existing ones to changes in acquisition parameters and image quality (Jrgen and Cristian, 2016).

An important drawback in using GAs is computation time. While genetic algorithms can be easily adapted to parallel or distributed architectures, it is computationally expensive to assess the performance of an image processing pipeline on a large image database of large three-dimensional images, such as breast tomosynthesis images. In our case study, average computation time for the saliency map ranged between 45 seconds and 4.5 minutes (average 2.5 minutes), depending on the number of scales. Even with a small dataset of 16 images, calculating the fitness function for the training set employs an average of 40 minutes, for a single genome, or 66 hours, for a single generation.

To address this limitation, we sought to reduce the search space not only by defining lower and upper bounds for each parameter, but also by using variable discretization steps. Arguably, setting up lower and upper bounds may require some manual experiments, although in our case lower and upper bounds for the scale parameters were established based on the average size of target lesions, while optimizing for smaller ones (i.e. $< 10mm$). We also performed a second experiment with relaxed lower and upper bounds, and obtained comparable results. Discretization effectively limits the search space to that of a grid search; as the same solution may need to be re-evaluated several times, results from previous computation can be stored and easily accessed. An important limitation of this study is that discretiza-

tion steps were manually selected based on prior knowledge. A more effective discretization strategy could optimize sampling of the parameter space, e.g. by employing non-uniform strategies, as well as reduce the time needed to apply the same framework to a different problem (Cantú-Paz, 2001).

One of the main limitations of this work lies in the selection of a very small dataset for the genetic algorithm implementation. While results generalized well to a larger dataset in terms of sensitivity, a larger dataset would make the training procedure more robust. The size of the dataset was chosen based on a trade-off between the calculation time required to evaluated the fitness function, and the complexity of the pipeline being trained. In our case, the pipeline was simple with a small number of parameters, but in the case of a more complex pipeline (e.g. with more processing steps, or machine learning-based) a much larger dataset would be needed. Besides increasing computational resources through parallel implementation, a more efficient implementation would be to evaluate the performance of a given individual incrementally, with lower degree of confidence, on smaller subsets of the database.

An interesting feature of genetic algorithms is that they enable an effective exploration of the most interesting regions of the parameter space. In our case study, viable solutions had average accuracy, calculated according to the modified Dice coefficient, ranging from roughly 0.4 to roughly 0.7. Exploratory data analysis techniques such as correlation and association rules were used to assess the effect of each parameter with the segmentation outcomes. Overall, the rules indicated segmentation accuracy was maximized when the values of M, m, T_s and Δs lie in the intervals $[12.85--18.5]$, $[3-5.333]$, $[0.92-0.937]$ and $[0-1.75]$. Among all individual parameters, M and T_g had the strongest impact on all outcome measures, while the correlation of m and Δs was negligible. Extracted association rule show a co-dependence of the values of M, Δs , T_s and T_g , indicating that it is probably best to adjust all parameters simultaneously to account for their inter-relationships. These considerations can be of practical use when adapting an existing pipeline to different datasets, vendor or acquisition parameters; further optimization of the algorithm can be improved by adjusting the search space accordingly, e.g. by adjusting discretization steps based on the strength of the association between segmentation quality and each individual parameter.

7. Conclusion

In conclusion, we proposed an evolutionary approach to the problem of finding optimal parameter values for segmentation pipelines to be used in the context of Computer Aided Detection pipelines. The framework can be applied to different algorithms with minimal prior knowledge of the parameter space, and provides an effective answer to the problem of fast, accurate, reliable and unsupervised training of segmentation pipelines to different scenarios. Knowledge derived from the parameter space can be use to further guide algorithm improvement or adaption to changes in the acquisition parameters, and to evaluate algorithm stability.

Further research is needed to prove the practicality of this approach to tune more complex detection and segmentation algorithms, especially based on deep learning. Ensuring a faster

and robust convergence of the GA, for instance through the use of adaptive, non-uniform parameter sampling, could allow to increase the size of the training set.

8. Acknowledgement

The authors wish to thank Azienda Ospedaliero Universitaria Cittá della Salute e della Scienza, Molinette, Turin, for providing the cases for this study.

References

Agliozzo, S., De Luca, M., Bracco, C., Vignati, A., Giannini, V., Martincich, L., Carbonaro, L., Bert, A., Sardanelli, F., Regge, D., 2012. Computer-aided diagnosis for dynamic contrast-enhanced breast mri of mass-like lesions using a multiparametric model combining a selection of morphological, kinetic, and spatiotemporal features. Medical physics 39 (4), 1704–1715.

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In: VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile. pp. 487–499.

Albelwi, S., Mahmood, A., 2016. Automated optimal architecture of deep convolutional neural networks for image recognition. In: Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on. IEEE, pp. 53–60.

Anastasio, M. A., Yoshida, H., Nagel, R., Nishikawa, R. M., et al., 1998. A genetic algorithm-based method for optimizing the performance of a computer-aided diagnosis scheme for detection of clustered microcalcifications in mammograms. Medical physics 25 (9), 1613–1620.

Angelie, E., de Koning, P., Danilouchkine, M., Van Assen, H., Koning, G., Van Der Geest, R., Reiber, J., 2005. Optimizing the automatic segmentation of the left ventricle in magnetic resonance images. Medical physics 32 (2), 369–375.

Apiletti, D., Baralis, E., Cerquitelli, T., Chiusano, S., Grimaudo, L., 2013. Searum: A cloud-based service for association rule mining. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013 / 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA-13 / 12th IEEE International Conference on Ubiquitous Computing and Communications, IUCC-2013, Melbourne, Australia, July 16-18, 2013. pp. 1283–1290.

Baralis, E., Cagliero, L., Cerquitelli, T., 2016. Supporting stock trading in multiple foreign markets: a multilingual news summarization approach. In: Proceedings of the Second International Workshop on Data Science for Macro-Modeling, DSMM@SIGMOD 2016, San Francisco, CA, USA, June 26 - July 1, 2016. pp. 3:1–3:6.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

Cantú-Paz, E., 2001. Supervised and unsupervised discretization methods for evolutionary algorithms. In: Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001). pp. 213–216.

Cerquitelli, T., Baralis, E., Morra, L., Chiusano, S., 2016. Data mining for better healthcare: A path towards automated data analysis? In: 32nd IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2016, Helsinki, Finland, May 16-20, 2016. pp. 60–63.

Corso, E. D., Cerquitelli, T., Ventura, F., 2017. Self-tuning techniques for large scale cluster analysis on textual data collections. In: Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017. pp. 771–776.

Dasgupta, D., Michalewicz, Z., 2013. Evolutionary algorithms – an overview. In: Evolutionary algorithms in engineering applications. Springer.

Delsanto, S., Griffa, M., Morra, L., 2006. Inverse problems and genetic algorithms. In: Universality of Nonclassical Nonlinearity. Springer, pp. 349–366.

Delsanto, S., Morra, L., Agliozzo, S., Baggio, R., Campanella, D., Tartaglia, V., Cerri, F., Iafrate, F., Neri, E., Laghi, A., et al., 2008. Computer aided detection of polyps in virtual colonoscopy with sameday faecal tagging. In: Medical Imaging. International Society for Optics and Photonics, pp. 69150T–69150T.

Delsanto, S., Morra, L., Griffa, M., Demartini, C., 2004. A genetic algorithms' approach to the exploration of parameter space in mesoscopic multicellular tumour spheroid models. In: Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE. Vol. 1. IEEE, pp. 675–678.

Dewancker, I., McCourt, M., Clark, S., 2016a. Bayesian optimization for machine learning: A practical guidebook. arXiv preprint arXiv:1612.04858.

Dewancker, I., McCourt, M., Clark, S., Hayes, P., Johnson, A., Ke, G., 2016b. A stratified analysis of bayesian optimization methods. arXiv preprint arXiv:1603.09441.

Diciotti, S., Picozzi, G., Falchini, M., Mascalchi, M., Villari, N., Valli, G., 2008. 3-d segmentation algorithm of small lung nodules in spiral ct images. IEEE transactions on Information Technology in Biomedicine 12 (1), 7–19.

Doi, K., 2007. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. Computerized Medical Imaging and Graphics 31 (45), 198 – 211, computer-aided Diagnosis (CAD) and Image-guided Decision Support.

Giannini, V., Vignati, A., De Luca, M., Agliozzo, S., Bert, A., Morra, L., Persano, D., Molinari, F., Regge, D., 2013. Registration, lesion detection, and discrimination for breast dynamic contrast-enhanced magnetic resonance imaging. In: Multimodality Breast Imaging: Diagnosis and Treatment. SPIE.

Goldberg, D. E., Holland, J. H., 1988. Genetic algorithms and machine learning. Vol. 3. Springer.

Greenspan, H., van Ginneken, B., Summers, R. M., 2016. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. IEEE Transactions on Medical Imaging 35 (5), 1153–1159.

Held, C., Nattkemper, T., Palmisano, R., Wittenberg, T., 2013. Approaches to automatic parameter fitting in a microscopy image segmentation pipeline: An exploratory parameter space analysis. Journal of pathology informatics 4 (Suppl).

Houssami, N., 2015. Digital breast tomosynthesis (3d-mammography) screening: data and implications for population screening.

Huang, C.-L., Wang, C.-J., 2006. A ga-based feature selection and parameters optimizationfor support vector machines. Expert Systems with applications 31 (2), 231–240.

Jrgen, W., Cristian, L., 2016. Four challenges in medical image analysis from an industrial perspective. Medical Image Analysis 33, 44 – 49, 20th anniversary of the Medical Image Analysis journal (MedIA).

Kuo, H.-C., Giger, M. L., Reiser, I., Drukker, K., Boone, J. M., Lindfors, K. K., Yang, K., Edwards, A., 2014. Impact of lesion segmentation metrics on computer-aided diagnosis/detection in breast computed tomography. Journal of Medical Imaging 1 (3), 031012–031012.

Lau, B., 2011. Optimization of breast tomosynthesis imaging systems for computer-aided detection. Tech. rep., CHICAGO UNIV IL.

Lee, J., Nishikawa, R. M., Reiser, I., Boone, J. M., 2017. Optimal reconstruction and quantitative image features for computer-aided diagnosis tools for breast ct. Medical Physics 44 (5), 1846–1856.

Li, Q., Nishikawa, R. M., 2015. Computer-aided detection and diagnosis in medical imaging. CRC Press.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., Sánchez, C. I., 2017. A survey on deep learning in medical image analysis. arXiv preprint arXiv:1702.05747.

Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Navruzyan, A., Duffy, N., Hodjat, B., 2017. Evolving deep neural networks. arXiv preprint arXiv:1703.00548.

Montemurro, M., Vincenti, A., Vannucci, P., 2013. The automatic dynamic penalisation method (adp) for handling constraints with genetic algorithms. Computer Methods in Applied Mechanics and Engineering 256, 70–87.

Morra, L., Sacchetto, D., Durando, M., Agliozzo, S., Carbonaro, A. L., Delsanto, S., Pesce, B., Persano, D., Mariscotti, G., Marra, V., Fonio, P., Bert, A., Oct 2015. Breast cancer: Computer-aided detection with

digital breast tomosynthesis. Radiology 277, 56–63.

Nemoto, M., Hayashi, N., Hanaoka, S., Nomura, Y., Miki, S., Yoshikawa, T., 2017. Feasibility study of a generalized framework for developing computer-aided detection systemsa new paradigm. Journal of Digital Imaging 30, 629–639.

Petrick, N., Sahiner, B., Armato, S. G., Bert, A., Correale, L., Delsanto, S., Freedman, M. T., Fryd, D., Gur, D., Hadjiiski, L., Huo, Z., Jiang, Y., Morra, L., Paquerault, S., Raykar, V., Samuelson, F., Summers, R. M., Tourassi, G., Yoshida, H., Zheng, B., Zhou, C., Chan, H.-P., 2013. Evaluation of computer-aided detection and diagnosis systemsa). Medical Physics 40 (8), 087001–n/a, 087001.

RapidMiner, Accessed: December 2015. R. m. p. the rapid miner project for machine learning, `http://rapid--i.com/`.

Sahiner, B., Chan, H., Petrick, N., 2002. Genetic algorithms for feature selection in computer-aided diagnosis. In: Computational Intelligence Processing in Medical Diagnosis. Springer, pp. 427–484.

Sahiner, B., Chan, H.-P., Petrick, N., Wagner, R. F., Hadjiiski, L., 2000. Feature selection and classifier performance in computer-aided diagnosis: The effect of finite sample size. Medical Physics 27 (7), 1509–1522.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R. M., 2016. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging 35 (5), 1285–1298.

Stanley, K. O., Miikkulainen, R., Jun. 2002. Evolving neural networks through augmenting topologies. Evol. Comput. 10 (2), 99–127.
URL `http://dx.doi.org/10.1162/106365602320169811`

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., Liang, J., 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE transactions on medical imaging 35 (5), 1299–1312.

Teare, P., Fishman, M., Benzaquen, O., Toledano, E., Elnekave, E., 2017. Malignancy detection on mammography using dual deep convolutional neural networks and genetically discovered false color input enhancement. Journal of digital imaging 30 (4), 499–505.

Teodoro, G., Kurç, T. M., Taveira, L. F., Melo, A. C., Gao, Y., Kong, J., Saltz, J. H., 2016. Algorithm sensitivity analysis and parameter tuning for tissue image segmentation pipelines. Bioinformatics 33 (7), 1064–1072.

van Ginneken, B., Schaefer-Prokop, C. M., Prokop, M., 2011. Computer-aided diagnosis: how to move from the laboratory to the clinic. Radiology 261 (3), 719–732.

Wang, J., Clark, S. C., Liu, E., Frazier, P. I., 2016. Parallel bayesian global optimization of expensive functions. arXiv preprint arXiv:1602.05149.

Wang, J., Nishikawa, R. M., Yang, Y., 2017. Global detection approach for clustered microcalcifications in mammograms using a deep learning network. Journal of Medical Imaging 4 (2), 024501–024501.

Zou, K. H., Warfield, S. K., Bharatha, A., Tempany, C. M., Kaus, M. R., Haker, S. J., Wells, W. M., Jolesz, F. A., Kikinis, R., 2004. Statistical validation of image segmentation quality based on a spatial overlap index. Academic radiology 11 (2), 178–189.