Efficient Parameter Optimization of Ensembles in Medical Image Analysis

egyetemi doktori (PhD) értekezés

a szerző neve: Tóth János
a témavezető neve: Dr. Hajdu András

Debreceni Egyetem
Természettudományi és Informatikai Doktori Tanács
Informatikai Tudományok Doktori Iskola
Debrecen, 2022

Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Informatikai Tudományok Doktori Iskola Diszkrét matematika, képfeldolgozás és komputergeometria programja keretében készítettem a Debreceni Egyetem műszaki doktori (PhD) fokozatának elnyerése céljából.
Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2022. ....

.....
a jelölt aláírása

Tanúsítom, hogy Tóth János doktorjelölt 2014-2018 között a fent megnevezett doktori iskola Diszkrét matematika, képfeldolgozás és komputergeometria programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult.
Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.
Az értekezés elfogadását javasolom.

Debrecen, 2022. ....

.....
a témavezető aláírása

Efficient Parameter Optimization of Ensembles in Medical Image Analysis

Értekezés a doktori (PhD) fokozat megszerzése érdekében
az informatika tudományágban

Írta: Tóth János okleveles programtervező matematikus

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskolája
Diszkrét matematika, képfeldolgozás és komputergeometria programja keretében

Témavezető: Dr. Hajdu András

A doktori szigorlati bizottság:

elnök:	Dr. Halász Gábor József	.....
tagok:	Dr. Ispány Márton	.....
	Dr. Palágyi Kálmán	.....

A doktori szigorlat időpontja: 2020. október 16.

Az értekezés bírálói:

	Dr.	.....
	Dr.	.....
	Dr.	.....

A bírálóbizottság:

elnök:	Dr.	.....
tagok:	Dr.	.....
	Dr.	.....
	Dr.	.....
	Dr.	.....

Az értekezés védésének időpontja: 2022. ....

Acknowledgement

I would like to thank my supervisor, Prof. Dr. András Hajdu for his valuable advice, guidance and support throughout my PhD studies.

I would also like to express my gratitude to my parents. It would not have been possible for me to complete my dissertation without their endless love, understanding and encouragement over the years.

This work is dedicated to my uncle, József.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In modern medicine, imaging has become a crucial tool. Technologies such as X-ray, ultrasound, computed tomography, magnetic resonance imaging, positron emission tomography, and medical photography are used extensively in clinical practice to confirm, assess, and document the course of numerous diseases and to evaluate the response to treatment.

Today, medical images – along with various omics (e.g., genomics and proteomics) data – make up the majority of data that needs to be processed and analyzed in healthcare. However, the manual examination of the acquired images is a labor-intensive process and subject to human error. In addition, the performance of an observer may depend on factors such as the level of experience, reading strategy, or fatigue. Therefore, as the number of imaging examinations started to increase, the demand for reliable automated methods to assist physicians also started to grow.

This need has given rise to the interdisciplinary research field of Medical Image Analysis that relies on image processing, pattern recognition, machine learning, and medicine to develop methods to extract

clinically relevant information from medical images in a reproducible and objective manner.

Because the procedures related to medical diagnosis are critical, it is usually not possible to rely on individual algorithms for their implementation. While a single algorithm may work well in general, it may fail in more difficult or infrequently encountered situations. For example, a lesion segmentation algorithm may perform differently on images acquired with different settings, which is a major problem when images from heterogeneous sources, e.g., from multiple sites, need to be processed. Or in the case of classification, the model of an algorithm may not be general enough to correctly classify all unknown instances, e.g., to classify all segmented lesion candidates.

To address these shortcomings, an ensemble of algorithms is often considered (see, e.g., [1–5]). Ensembles are constructed from such algorithms (members) that are based on different principles, models, etc. to solve a specific problem [6]. The diversity of the members allows the ensemble to respond more flexibly to various conditions [7]. The basic idea of the ensemble methodology is that by combining the outputs of multiple algorithms using an appropriate aggregation rule [8, 9], a system can be created that outperforms each of its constituent members if certain conditions on their diversity and individual performance are met [10, 11].

A critical issue with ensembles is that using the individually optimal parameter setting of the members may not necessarily maximize the performance of the ensemble itself. For this reason, parameter optimization at the ensemble level is required, which can lead to a large-scale problem depending on the number and range of the parameters of the members. Even if the individual members have only a few parameters that can take values from limited ranges, the search space of the possible parameter settings of the ensemble can still be

very large. Accordingly, using exhaustive search to find the optimal parameter setting quickly becomes impractical.

To solve such large-scale combinatorial optimization problems, stochastic search methods are commonly used. These methods can be divided into instance-based and model-based ones. The main difference between these two groups is that instance-based methods, such as simulated annealing (SA) [12] and the genetic algorithms [13], generate new candidate solutions based on the current solution(s), while model-based methods, such as ant colony optimization [13] and the cross-entropy method [14], generate candidate solutions through more expensive, adaptive stochastic mechanisms.

Stochastic approaches can efficiently find good solutions to large-scale problems by sacrificing some accuracy for a substantial reduction in search cost. However, even a stochastic search can be very expensive if the evaluation of a solution is itself expensive, e.g., due to the high complexity of the objective function or the large size of the dataset, which latter is often necessary to avoid parameter overfitting.

One way to reduce the cost of stochastic optimization is to use partial data at each iteration, i.e., to approximate the value of the objective function instead of determining it exactly. A similar principle is applied in the stochastic gradient descent (SGD) [15] and mini-batch gradient descent (MGD) [16] algorithms, which are widely used in machine learning tasks.

SGD is a method for the optimization of differentiable objective functions. Unlike the classic gradient descent (GD) method, which computes the gradient at each iteration using the entire dataset, SGD estimates the gradient using a single random example from the dataset. This approach reduces the cost per iteration at the expense of a lower convergence rate. While SGD generally progresses quickly at the beginning, the variance of the gradient estimates deteriorates the conver-

gence rate as the optimum is approached [17]. That is, SGD requires more iterations to converge. However, for large datasets, the overall cost is still likely to be less than in the case of GD.

MGD estimates the gradient at each iteration as an average with respect to a random subset (mini-batch) of the dataset. It has been shown to be an efficient optimization method for large-scale machine learning problems, as it reduces the cost of optimization compared to GD while providing solutions that are usually superior to those produced by SGD considering the same overall cost. MGD uses a constant batch size, typically determined by empirical analysis. The batch size has a significant impact on the convergence of MGD, as it affects the variance of the gradient estimates. Combined with back-propagation, MGD is the most commonly used method for training deep neural networks [18]. In these applications, the batch size of MGD is usually small (e.g., 32, 64, or 128), which is expected to lead to better generalization ability due to the noise of the gradient estimates [19]. In recent years, however, the use of large batch sizes in MGD has also been actively investigated [20–22] to reduce the number of parameter updates required during training.

Note that in both SGD and MGD, the gradient estimates become noisier as the solutions approach the optimum. For this reason, convergence is facilitated by a decreasing learning rate [23], which results in an increased overall cost of optimization.

In our preliminary studies [24, 25], we have successfully applied an approach similar to that of MGD to reduce the cost of parameter optimization of image processing ensembles by using partial data in each iteration. For our studies, we chose the metaheuristic SA, which is widely used to solve both discrete and continuous optimization problems due to its simplicity and appealing properties. Namely, we proposed methods based on SA that evaluate the objective function using

a subset of the dataset obtained at each iteration by random sampling. First, the effect of using fixed-size samples of the dataset at each iteration was investigated using the parameter optimization problem of a lesion detector ensemble. It was found that even for more complex objective functions, it is possible to find a fixed sample size that leads to good solutions. It was also confirmed that the considered lesion detector ensemble performs better when using the parameters obtained with ensemble-level optimization than when using the individually optimal parameters of the members. However, the determination of the sample size required to obtain good solutions was heuristic and problem dependent. We also found that more optimization runs were required due to the uneven quality of the achievable solutions caused by the noise of the objective function estimates originating from the sampling. Therefore, we also tested a simple scheduled increase in the sample size during the search, which improved the average quality of the solutions. The above preliminary results led us to the conclusion that to effectively use reduced data for parameter optimization of ensembles with SA, the noise must be controlled during the search to maintain convergence of the method.

The goal of this research was to develop methods for the efficient parameter optimization of ensembles performing medical image analysis tasks. In this dissertation, three stochastic methods are proposed for this purpose, all of which are based on SA and use noisy evaluation in different ways to reduce the overall cost of the search. Different approaches are presented to approximate the value of the objective function with partial training data to evaluate solutions, i.e., the performance of the ensemble at a given parameter setting. Of course, the noise introduced by using partial training data may cause the search method to consider an inferior solution as superior due to the inaccurate value of the objective function and vice versa. For this reason,

appropriate strategies have been developed for each method to control the noise during the search process by the amount of data used for evaluation in order to maintain the achievable solution quality.

The main contributions of this dissertation can be summarized as follows (including the related publications):

- Optimization with dataset sampling

  1. An efficient stochastic method is proposed for the parameter optimization of ensembles on large training sets that uses sampling-based objective function evaluation [P4, P8, P14, P16, P18].
  2. A closed-form equation is given to determine the minimum sample size required to evaluate a solution at a given iteration in order to maintain the convergence of the method in probability [P4, P16, P18].

- Optimization with image downscaling

  3. An efficient stochastic method is proposed for the parameter optimization of image processing (segmentation) ensembles that uses increasingly higher resolution levels of a pyramid representation of the images in the training set to evaluate the objective function during the search [P4, P11].
  4. A strategy is proposed to select the highest scaling level that can be used to evaluate a solution at a given iteration in order to maintain the convergence of the method in probability for a given pyramid representation of the images. [P4, P11].

- Optimization with combined noisy evaluation

  5. A stochastic method is proposed for accelerating the parameter optimization of image processing (segmentation) ensembles that combines dataset sampling with image downscaling for the evaluation of the objective function [P1, P4, P11].

The rest of this dissertation is organized as follows. Chapter 2 describes the theoretical background of the research conducted. First, SA is briefly introduced, focusing on its convergence properties in the case of using imprecise measurements to evaluate solutions and the design choices necessary for its implementation. Then, the basic concepts and notations used in this dissertation are defined. In Chapter 3, a method is presented for the parameter optimization of ensembles on large (image) datasets. This method accelerates the search by evaluating the solutions using subsets of the dataset obtained by random sampling. As another approach to accelerating parameter optimization, Chapter 4 presents a method that uses increasingly higher resolution levels of a pyramid representation of the dataset images to evaluate solutions during the optimization process. In Chapter 5, a method that combines the previous two approaches is discussed. It is shown that by using optimal combinations of sample size and scaling level, the search can be further accelerated when the cardinality of the dataset is below a problem-specific value.

Chapter 2

Background

2.1 Simulated annealing

SA is a local search algorithm that was introduced by Kirkpatrick *et al.* [12] and independently by Černý [26] to address difficult combinatorial optimization problems. Simulated annealing is inspired by the physical annealing process in metallurgy, in which a metal is heated and then slowly cooled until it reaches its lowest lattice energy state and is thus free of crystal defects. If cooling is sufficiently slow, the final configuration results in a metal with improved structural integrity.

The main feature of SA is the capacity to escape from local optima by accepting non-improving moves with a probability that depends on the difference in the objective function (energy) values between the current and candidate states, and a decreasing control parameter (temperature). The method applied to generate the sequence of temperature levels is called a cooling schedule, and its choice strongly influences the performance of SA.

The simplicity and general applicability of SA have resulted in this procedure being used widely to address both discrete and continuous

optimization problems. For a comprehensive discussion of the theory and application of SA, see [27].

2.1.1 Implementation of SA

The general operation of SA is represented in Algorithm 1.

Algorithm 1 Simulated Annealing

Input: Initial state π_{init}
Initial temperature $T^{(0)}$

```
1: k ← 0
2: π ← π_init
3: E ← CALCULATE_ENERGY(π)
4: while OUTER-LOOP CRITERION SATISFIED do
5:    while INNER-LOOP CRITERION SATISFIED do
6:       π_cand ← GENERATE_NEIGHBOR(π)
7:       E_cand ← CALCULATE_ENERGY(π_cand)
8:       r ← RAND([0, 1])
9:       if ACCEPT(E, E_cand, T^(k), r) then
10:         π ← π_cand
11:         E ← E_cand
12:      end if
13:   end while
14:   T^(k+1) ← UPDATE_TEMPERATURE(T^(0), k)
15:   k ← k + 1
16: end while
17: return π
```

As it can be seen, a number of design choices must be made to implement SA. In particular, we have to specify the energy function, the neighborhood function, the acceptance criterion, the cooling schedule, the thermal equilibrium criterion, and the termination criterion:

- *Input – Initial state π_{init}*: The search starts from the initial state π_{init}, which is typically selected at random from the search space.

- *Input – Initial value of the control parameter $T^{(0)}$*: The initial temperature $T^{(0)}$ should be determined to allow virtually all state transitions to be accepted. Kirkpatrick *et al.* [12] suggested that a suitable value should result in an initial acceptance probability χ_0 of about 0.8. For instance, using the acceptance criterion defined by (2.3), we can calculate $T^{(0)}$ as

$$T^{(0)} = -\frac{\Delta E_{max}}{\ln(\chi_0)}, \tag{2.1}$$

  where ΔE_{max} is the maximum possible energy difference between any two states.

- *Lines 3 and 7 – Energy function* CALCULATE_ENERGY: The objective function of the optimization, typically formulated to suit the minimization approach of SA. That is, the algorithm seeks a solution with minimal energy.

- *Line 4 – Termination criterion* OUTER-LOOP CRITERION: When the temperature falls below the final value T_{final}, the search is stopped. At the final temperature T_{final}, the acceptance probability χ_{final} should be almost 0. In a similar manner to the initial temperature, the final temperature is calculated as:

$$T_{final} = -\frac{\Delta E_{min}}{\ln(\chi_{k_{final}})}, \tag{2.2}$$

  where ΔE_{min} is the minimum possible non-zero energy difference between any two states.

- *Line 5 – Thermal equilibrium criterion* INNER-LOOP CRITERION: Each iteration of the inner loop generates a new state that is ac-

cepted or rejected depending on the function ACCEPT. The loop ends when "thermal equilibrium" is reached, i.e., when the INNER-LOOP CRITERION is satisfied. This criterion usually uses either a maximum number of states to be generated, a maximum number of acceptances, or a combination of the two. Implementations often omit this criterion and execute the inner loop statements once.

- *Line 6 – Neighborhood function* GENERATE_NEIGHBOR: The method of generating a neighbor state depends strongly on the optimization problem. For black-box energy functions, a general approach is to randomly select states from a neighborhood whose size decreases as the search progresses. Note that the algorithm explores the state space by random sampling when the neighborhood is large, while it focuses on specific regions when the neighborhood is small.

- *Line 9 – Acceptance function* ACCEPT: The acceptance function decides whether a move from the state π to π_{cand} is accepted based on the probability $\chi_{\pi,\pi_{cand}}$, which is determined by an acceptance criterion using the current temperature $T^{(k)}$, the energy E of the current state, and the energy E_{cand} of the candidate state. A move is accepted if $\chi_{\pi,\pi_{cand}}$ is greater than a uniform random number $r \in [0, 1)$ generated by the function RAND. The most commonly used acceptance criteria are the following.

  - Metropolis criterion:

$$\chi_{\pi,\pi_{cand}} = \begin{cases} \exp\left(\frac{E-E_{cand}}{T^{(k)}}\right), & \text{if } E_{cand} > E, \\ 1, & \text{otherwise.} \end{cases} \tag{2.3}$$

When using the Metropolis criterion, the acceptance probability of a move to a non-inferior candidate state is 1. However, to avoid being stuck in local optima, moves to inferior candidate states may also be

accepted. To this end, $\exp\left((E - E_{cand})/T^{(k)}\right)$ gives the acceptance probability of the move, which decreases as the temperature $T^{(k)}$ decreases.

- Barker criterion:

$$\chi_{\pi,\pi_{cand}} = \frac{1}{1 + \exp\left(\frac{E_{cand}-E}{T^{(k)}}\right)}. \tag{2.4}$$

When using the Barker criterion, even superior states may be rejected if they do not significantly improve the energy. As the temperature $T^{(k)}$ decreases, superior states are more likely to be accepted unless the energy difference is negligible, while inferior states are more likely to be rejected, as when using (2.3).

- *Line 14 – Temperature function* UPDATE_TEMPERATURE: One of the most important design decisions in the implementation of the algorithm is the selection of an appropriate temperature function. When the time budget of the search is limited, using a slow cooling schedule will result in a failed search, while using a schedule that is too fast may result in the search being stuck in a local optimum. The temperature function together with the initial temperature $T^{(0)}$ form a cooling schedule that is used to systematically decrease the temperature, thus decreasing the probability of accepting moves to states with worse energy values. The most commonly used temperature functions are the following.

  - Exponential temperature function:

$$T^{(k)} = T^{(0)}\,\alpha^k \text{ with } 0 < \alpha < 1. \tag{2.5}$$

- Linear temperature function:

$$T^{(k)} = T^{(0)} - \alpha k \text{ with } 0 < \alpha. \tag{2.6}$$

- Logarithmic temperature function:

$$T^{(k)} = \frac{T^{(0)}}{1 + \alpha \ln(1 + k)} \text{ with } 1 < \alpha. \tag{2.7}$$

Note that the logarithmic temperature function is based on the asymptotic convergence condition of SA [28], but includes the factor α that allows practical application of the schedule.

2.1.2 Convergence of SA in presence of noise

Originally, SA was designed based on the assumption that the energy of a state can be calculated exactly, but the evaluation of a state is often subject to noise in practical problems. As a consequence, a number of studies have investigated the convergence properties of SA in noisy environments.

The first study of this topic by Kushner [29] involved asymptotic analysis of SA under suitable conditions based on the theory of large deviations while assuming Gaussian noise.

By considering discrete search spaces and assuming that the noise is normally distributed with mean 0 and variance $\left(\sigma^{(k)}\right)^2 > 0$ in the k-th ($k \in \mathbb{N}$) iteration, Gelfand and Mitter proved [30] that SA converges to the global optimum in probability using noisy evaluation in the same manner as using exact energy values if the standard deviation $\sigma_d^{(k)}$ of the noise in the k-th iteration for each k is dominated by the

temperature $T^{(k)}$, i.e., when

$$\sigma_d^{(k)} = o\left(T^{(k)}\right), \tag{2.8}$$

where o is a Bachmann–Landau symbol that expresses a stronger requirement on the asymptotic behavior of a function than O (for further details, see [31]).

Assuming the same noise properties for a specific annealing schedule, Gutjahr and Pflug [32] proved that SA converges in probability to the globally optimal solution if the standard deviation of the noise is at least inversely proportional to the number of iterations, i.e., when

$$\sigma^{(k)} = O\left(k^{-\gamma}\right) \text{ with some } \gamma > 1. \tag{2.9}$$

They generalized the proof of convergence to an arbitrary noise distribution that is symmetric and more peaked around 0 than the Gaussian distribution.

2.2 Basic concepts and notations

This section introduces the basic concepts and notations used in this dissertation, which will be supplemented in subsequent chapters in relation to the methods discussed.

Let $\mathcal{D} = \{D_1, D_2, \ldots, D_M\}$ be an ensemble of $M \in \mathbb{N}$ member algorithms and Λ the set of images. The output of the algorithm D_i $(i = 1, 2, \ldots, M)$ is denoted by $D_i(\lambda)$ for an image $\lambda \in \Lambda$ and, where applicable, the pixel value of the output at the coordinates (x, y) by $D_i(\lambda)_{(x,y)}$. The output $\mathcal{D}(\lambda)$ of the ensemble for λ is determined by applying an aggregation rule to the individual outputs of the algorithms $D_1, D_2, \ldots, D_M$.

In the case of a classification problem, let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_J\}$ be a set of finite class labels. The classifier D_i assigns the support values $D_i(\lambda) = (d_{i,1}(\lambda), \ldots, d_{i,J}(\lambda))$ to λ, which describes the opinion of the classifier in terms of the degree to which λ should be labeled by $\omega_1, \ldots, \omega_J$, respectively.

The simple majority voting-based ensemble classifier can be derived by restricting the support of the individual classifiers with $d_{i,j}(\lambda) = \delta_{rj}$, where $j = 1, 2, \ldots, J$ if the classifier D_i assigns the class ω_r to λ. The final labeling by the ensemble is based on determining the class that receives the largest support in terms of the number of votes.

Different parameter settings can be considered for the member algorithms, so we let Π_i denote the parameter domain and $\pi_i \in \Pi_i$ a given parameter of the algorithm D_i $(i = 1, 2, \ldots, M)$. Furthermore, let $\pi \in \Pi = \Pi_1 \times \Pi_2 \times \ldots \times \Pi_M$ denote a given parameter vector of the ensemble. Then, the ensemble with a specific parameter setting π will be denoted by $\mathcal{D}^{(\pi)}$.

In our applications, $\lambda \in \Lambda_N \subset \Lambda$, where Λ_N is a set of $N \in \mathbb{N}$ images, and the ensemble members are medical image analysis algorithms, whose outputs are aggregated using majority voting-based rules.

Chapter 3

Optimization with Dataset Sampling

3.1 Introduction

In this chapter, we present a method for accelerating parameter optimization of ensembles on large image datasets. Namely, we propose an efficient sampling-based evaluation method for SA that considers only the minimum required portion of the dataset in each iteration to accelerate the search while maintaining its convergence properties.

The development of this method was motivated by our previous research [24, 25], in which we successfully tested the evaluation of the objective function over only a certain subset of the dataset prepared in every search step by random sampling of the dataset images; however, our approach was heuristic regarding the level of sampling applied during the search process.

The main contribution of our approach is the correspondence of dataset sampling to the noisy evaluation of the objective function. The sample sizes required during the search process are theoretically

determined by adapting the convergence results for noisy evaluation in SA.

To assess the applicability of the method, we prepared and optimized two ensembles for diabetic retinopathy (DR) pre-screening based on microaneurysm (MA) detection with convolutional neural network-based and traditional object detectors. Our experimental results indicate that the proposed method substantially reduces the time required for the search without compromising the quality of the solution.

The remainder of this chapter is organized as follows. In Section 3.2, we describe our sampling strategy and give the SA-based search algorithm incorporating it. Our main result regarding the determination of the minimum sample size required during the search is also formulated in this section as Theorem 1. In Section 3.3, we present an application to retinal image analysis. Our experimental findings regarding the classification of retinal images according to DR are presented in Section 3.4. Detailed results are provided in terms of the computational time reductions obtained using the proposed method while also maintaining the solution quality. We also showed that an efficient ensemble of MA detectors can be prepared for pre-screening DR. Besides, we demonstrated that the proposed method can also be used to optimize the detector ensemble for the accurate localization of MAs. Finally, we present our conclusions in Section 3.5.

3.2 SA with sampling-based evaluation

Next, we describe an evaluation method for SA that maintains the quality of the achievable solution while reducing the runtime for energy functions commonly used to evaluate the average performance of object detectors and classifiers on datasets. For this, we propose

a sampling strategy that is based on the convergence results for the noisy evaluation of the energy function.

3.2.1 Noise originating from sampling

To consider noisy evaluation of the energy, the ensemble $\mathcal{D}^{(\pi)}$ with accuracy $p_{\mathcal{D}^{(\pi)}} \in [0,1]$ is a discrete random variable $X_{\mathcal{D}^{(\pi)}}$ with mean $\mathbb{E}(X_{\mathcal{D}^{(\pi)}})$ and variance $\mathrm{Var}(X_{\mathcal{D}^{(\pi)}})$, where $\mathbb{E}(X_{\mathcal{D}^{(\pi)}}) = p_{\mathcal{D}^{(\pi)}}$. Furthermore, let $x^i_{\mathcal{D}^{(\pi)}}$ denote the i-th realization of $X_{\mathcal{D}^{(\pi)}}$ $(i = 1, \ldots, N)$.

Definition 3.1. *The energy E_π used to evaluate the performance of the ensemble $\mathcal{D}^{(\pi)}$ for a given parameter setting π is determined as the empirical mean value of $X_{\mathcal{D}^{(\pi)}}$, i.e., the mean $\mu^N_{\mathcal{D}^{(\pi)}}$ of N realizations:*

$$E_\pi = \mu^N_{\mathcal{D}^{(\pi)}} = \frac{1}{N} \sum_{i=1}^{N} x^i_{\mathcal{D}^{(\pi)}}. \tag{3.1}$$

Calculating the energy can be computationally expensive when considering large populations, so we estimate it using sampling.

Definition 3.2. *Let Λ_n be a random sample of size n taken from the finite population Λ_N, i.e., $\Lambda_n \subseteq \Lambda_N$ $(0 < n \leq N)$. The energy estimate $\widehat{E}_{\Lambda_n,\pi}$ to estimate the performance of the ensemble $\mathcal{D}^{(\pi)}$ for a given parameter setting π using Λ_n is determined as the sample mean $\bar{x}^{\Lambda_n}_{\mathcal{D}^{(\pi)}}$:*

$$\widehat{E}_{\Lambda_n,\pi} = \bar{x}^{\Lambda_n}_{\mathcal{D}^{(\pi)}} = \frac{1}{n} \sum_{j:\lambda_j \in \Lambda_n} x^j_{\mathcal{D}^{(\pi)}}. \tag{3.2}$$

If the parameter setting π is fixed, then we use the brief notations E and $\widehat{E}_{\Lambda_n}$ instead of E_π and $\widehat{E}_{\Lambda_n,\pi}$, respectively.

As a special case, in a binary classification problem, the ensemble $\mathcal{D}^{(\pi)}$ with classification accuracy $p_{\mathcal{D}^{(\pi)}}$ is a random variable $X_{\mathcal{D}^{(\pi)}}$ from

a Bernoulli distribution with

$$P(X_{\mathcal{D}^{(\pi)}} = 1) = p_{\mathcal{D}^{(\pi)}}, \quad \text{and} \quad P(X_{\mathcal{D}^{(\pi)}} = 0) = 1 - p_{\mathcal{D}^{(\pi)}}, \tag{3.3}$$

where $X_{\mathcal{D}^{(\pi)}} = 1$ and $X_{\mathcal{D}^{(\pi)}} = 0$ denote correct and incorrect classification by $\mathcal{D}^{(\pi)}$, respectively. In this case, for the theoretical mean and variance of the variable $X_{\mathcal{D}^{(\pi)}}$ from a Bernoulli distribution, we have

$$\mathbb{E}(X_{\mathcal{D}^{(\pi)}}) = p_{\mathcal{D}^{(\pi)}}, \quad \text{and} \quad \mathrm{Var}(X_{\mathcal{D}^{(\pi)}}) = p_{\mathcal{D}^{(\pi)}}(1 - p_{\mathcal{D}^{(\pi)}}). \tag{3.4}$$

Assuming that calculating each value $x^i_{\mathcal{D}^{(\pi)}}$ $(i = 1, \ldots, N)$ has the same computational cost, then calculating $\widehat{E}_{\Lambda_n}$ is n/N times less computationally expensive than calculating E, but using $\widehat{E}_{\Lambda_n}$ introduces noise in the evaluation.

Definition 3.3. *For a sample Λ_n, the noise d_{Λ_n} originating from the sampling is determined as follows:*

$$d_{\Lambda_n} = \widehat{E}_{\Lambda_n} - E = \bar{x}^{\Lambda_n}_{\mathcal{D}^{(\pi)}} - \mu^N_{\mathcal{D}^{(\pi)}}. \tag{3.5}$$

3.2.2 Sampling strategy and its algorithmic realization

Because of the noisy evaluation of the energy function, SA may consider an inferior state to be superior and vice versa. That is, the noise makes the search more random and usually lowers the quality of the solution that can be reached after a given number of steps.

According to (2.8), to ensure the convergence of SA in the presence of noise, a sampling strategy must be applied that is suitable for controlling the standard deviation of the noise $\sigma_{d_{\Lambda_n}}$ regarding the temperature T during the search by selecting an appropriate sample size in each search step. Thus, we must determine the maximum allowed

value $\sigma_{d_n}^{(k)}$ of each $\sigma_{d_{\Lambda n}}$ for the current temperature $T^{(k)}$ in order to find the minimum sample size required. We state Lemma 1 for this purpose. Naturally, the standard deviation of the noise will be smaller when the sample size n is closer to the population size N.

Lemma 1. *A sufficiently simple general form of $\sigma_{d_n}^{(k)}$ that maximizes its value at the temperature $T^{(k)}$ can be given as follows:*

$$\sigma_{d_n}^{(k)} \gtrsim T^{(k)}(1-\epsilon)^k, \quad 0 < \epsilon \ll 1. \tag{3.6}$$

Proof. Derived from (2.8),

$$\lim_{k\to\infty} \frac{\sigma_{d_n}^{(k)}}{T^{(k)}} = 0 \tag{3.7}$$

must hold to preserve the convergence of the method in probability. To maintain the limit in (3.7), the sequence $\left\{\sigma_{d_n}^{(k)}\right\}$ has to be decreasing such that $\lim_{k\to\infty} \sigma_{d_n}^{(k)} = 0$ and $\sigma_{d_n}^{(k)} < T^{(k)}$ for each $k \in \mathbb{N}$. Based on these conditions, a sufficiently simple general form of $\sigma_{d_n}^{(k)}$ that maximizes its value can be given as (3.6). □

Example 1. As an application of Lemma 1, by considering the exponential cooling schedule with

$$T^{(k)} = T^{(0)}\,\alpha^k \text{ with } 0 \le \alpha \le 1, \tag{3.8}$$

the maximum value of $\sigma_{d_n}^{(k)}$ can be approximated as

$$\sigma_{d_n}^{(k)} \approx T^{(0)}\,\alpha^k(1-\epsilon)^k \text{ with } 0 \le \alpha \le 1, \text{ and } 0 < \epsilon < 1. \tag{3.9}$$

A similar derivation can be applied for other cooling schedules as well.

Now we can formulate our main theoretical contribution regarding how to determine the sample size during the search.

Theorem 1. *For an arbitrary cooling schedule, the minimum sample size $n^{(k)}$ required at the k-th iteration to maintain the convergence of the method in probability can be estimated as*

$$n^{(k)} \approx \frac{N\sigma_{max}^2}{(N-1)\sigma_{d_n}^{(k)^2} + \sigma_{max}^2}, \tag{3.10}$$

where σ_{max} is the worst-case, maximum value of the population standard deviation $\sigma_N^{\mathcal{D}(\pi)}$, and $\sigma_{d_n}^{(k)}$ can be derived using Lemma 1.

Proof. The noise defined in (3.5) is actually the difference between the sample mean and its expected value (the population mean), so its standard deviation is equal to the standard deviation of the sampling distribution of the mean, i.e., the standard error of the mean $\sigma_{\bar{x}_n^{\mathcal{D}(\pi)}}$. Therefore, the standard deviation of the noise can be calculated as follows:

$$\sigma_{d_n} = \sigma_{\bar{x}_n^{\mathcal{D}(\pi)}} = \frac{\sigma_N^{\mathcal{D}(\pi)}}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}}, \tag{3.11}$$

where $\sigma_N^{\mathcal{D}(\pi)}$ is the population standard deviation and $\sqrt{\frac{(N-n)}{(N-1)}}$ is the finite population correction factor.

In (3.11), the population standard deviation $\sigma_N^{\mathcal{D}(\pi)}$ is unknown, but it can be estimated using its worst-case (maximum) value σ_{max}. It should be noted that in this case, it is not possible to estimate the population standard deviation with the sample standard deviation because the required sample size is not yet known.

Using the maximum value of the population standard deviation, the minimum required sample size $n^{(k)}$ at the k-th iteration can be determined as (3.10). □

Example 2. For example, considering the exponential cooling schedule given in (3.8) and $\sigma_{max} = 0.5$, the minimum sample size in the k-th iteration can be given as

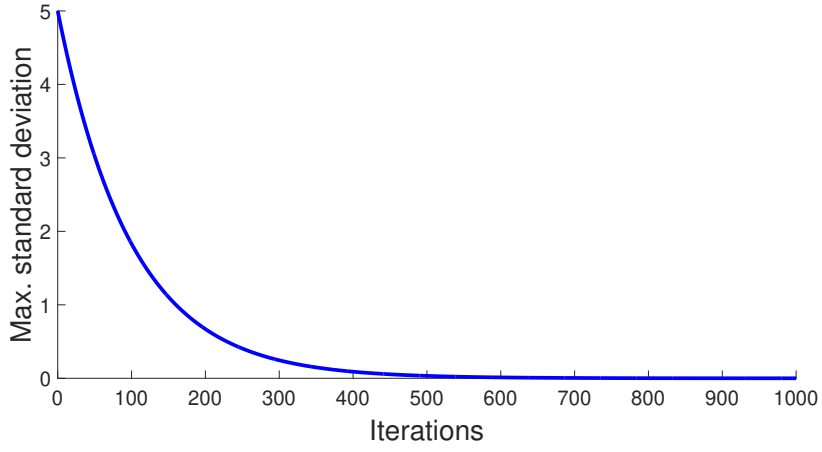
$$n^{(k)} = \frac{N}{4\,(N-1)\,(T^{(0)}\,\alpha^k(1-\epsilon)^k)^2 + 1}. \tag{3.12}$$

As a numeric demonstration for the example given above, let us consider $T^{(0)} = 5$, $k = 1000$, $\alpha = 0.99$, and $N = 2000$. For this setup, during the SA search, the maximum values allowed for the standard deviation of the noise $\sigma_{d_n}^{(k)}$ and the corresponding required sample sizes $n^{(k)}$ are shown in Fig. 3.1(a) and 3.1(b), respectively.

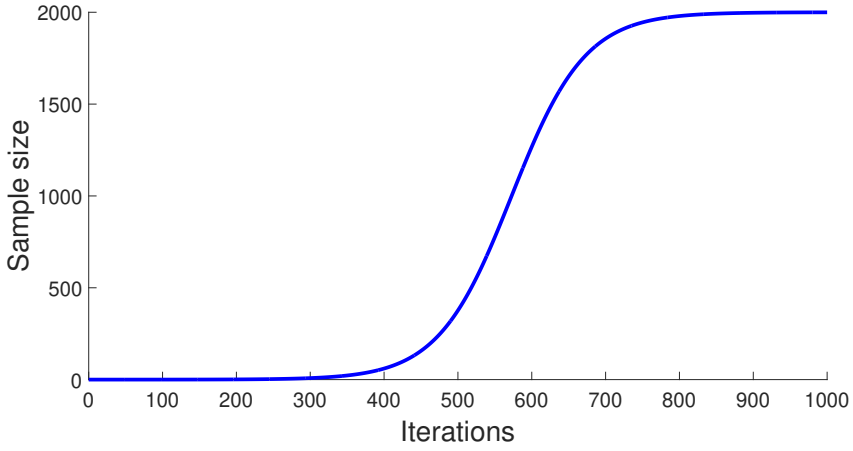
One technical issue should be noted: for every temperature value $T^{(k)}$, a minimum required sample size $n^{(k)}$ must be used; therefore, the energy estimate of the current state should be recomputed over a sufficiently large sample in every iteration, i.e., when the temperature decreases, in order to compare the quality of the current and candidate states. However, recomputing this value would be time consuming and the evaluation would become less effective than the complete evaluation after at least half of the population is included in the sample. Therefore, in each iteration, we normalize the energy estimate of the current state using the ratio of the minimum required sample sizes at the previous and current temperatures as

$$\widehat{E}_{norm} = \widehat{E}_{\Lambda_n^{(k-1)}} \cdot (n^{(k-1)}/n^{(k)}), \tag{3.13}$$

where $n^{(k-1)}$ is the sample size at which the energy estimate $\widehat{E}_{\Lambda_n^{(k-1)}}$ of the current state is calculated and $n^{(k)}$ is the sample size at which the energy of the neighbor state will be calculated. It should be noted that the factor used for correction becomes gradually less significant as the search proceeds. As a secondary technical issue, we consider



(a)



(b)

Figure 3.1. Example of the sampling strategy in SA search with an exponential cooling schedule: (a) maximum standard deviation of the noise and (b) minimum required sample size.

that minimum sample size should be $n \geq 50$ in order to make a reasonable assumption regarding the Gaussian distribution of the noise d_n by following the general recommendations (see [33].)

Our approach for finding the optimal parameter setting for an ensemble using the proposed sampling strategy is formally described in Algorithm 2. We refer to this algorithm as SA with Sampling-based Evaluation (SA-SBE) in the following. The algorithm contains several tunable parameters and functions, which must be selected according to the desired application. The setup corresponding to our object detection task is described in Section 3.3.3.

3.3 Application: DR pre-screening

DR is a complication of diabetes mellitus caused by progressive damage to the blood vessels in the retina, which is the light-sensitive lining in the back of the eye. DR is one of the leading causes of vision loss worldwide, but the risk of blindness can be significantly reduced through early diagnosis and timely treatment [34]. Therefore, patients with diabetes mellitus should undergo regular DR screening, but the manual grading of cases is resource-demanding and prone to human error. Consequently, over the last two decades, considerable efforts have been made to establish reliable automated methods to facilitate the mass screening of DR using color retinal photographs and various working principles, such as red and bright lesion detection [35, 36], feature extraction and classification [37,38], and deep learning [39,40].

3.3.1 DR screening based on MA detection

Several of the methods mentioned above aim to assign grades to input retinal images according to the severity of DR. However, even the

Algorithm 2 Simulated Annealing with Sampling-based Evaluation (SA-SBE)

Input: An ensemble $\mathcal{D} = \{D_1, \ldots, D_L\}$ with free parameters $\Pi = \Pi_1 \times \cdots \times \Pi_L$.
A population for classification Λ_N.
Maximum standard deviation σ_{max} of the energy.
Initial parameter setting $\pi_{init} \in \Pi$.
Initial temperature $T^{(0)}$.

$k \leftarrow 0$
$\pi \leftarrow \pi_{init}$
$n \leftarrow$ SAMPLE_SIZE$(T^{(0)}, k, \sigma_{max})$
$\Lambda_n \leftarrow$ TAKE_SAMPLE(Λ_N, n)
$\widehat{E}_{\Lambda_n,\pi} \leftarrow$ CALCULATE_ENERGY$(\pi, \Lambda_n, \mathcal{D})$
while OUTER-LOOP CRITERION SATISFIED **do**
  $n_{prev} \leftarrow n$
  $n \leftarrow$ SAMPLE_SIZE$(T^{(0)}, k, \sigma_{max})$
  $\widehat{E}_{\Lambda_n,\pi} \leftarrow$ ENERGY_NORMALIZATION$(\widehat{E}_{\Lambda_n,\pi}, n_{prev}, n)$
  while INNER-LOOP CRITERION SATISFIED **do**
    $\pi_{cand} \leftarrow$ GENERATE_NEIGHBOR(π)
    $\Lambda'_n \leftarrow$ TAKE_SAMPLE(Λ_N, n)
    $\widehat{E}_{\Lambda'_n,\pi_{cand}} \leftarrow$ CALCULATE_ENERGY$(\pi_{cand}, \Lambda'_n, \mathcal{D})$
    $r \leftarrow$ RAND$([0, 1])$
    if ACCEPT$(\widehat{E}_{\Lambda_n,\pi}, \widehat{E}_{\Lambda'_n,\pi_{cand}}, T^{(k)}, r)$ **then**
      $\pi \leftarrow \pi_{cand}$
      $\widehat{E}_{\Lambda_n,\pi} \leftarrow \widehat{E}_{\Lambda'_n,\pi_{cand}}$
    end if
  end while
  $T^{(k+1)} \leftarrow$ UPDATE_TEMPERATURE$(T^{(0)}, k)$
  $k \leftarrow k + 1$
end while
return π

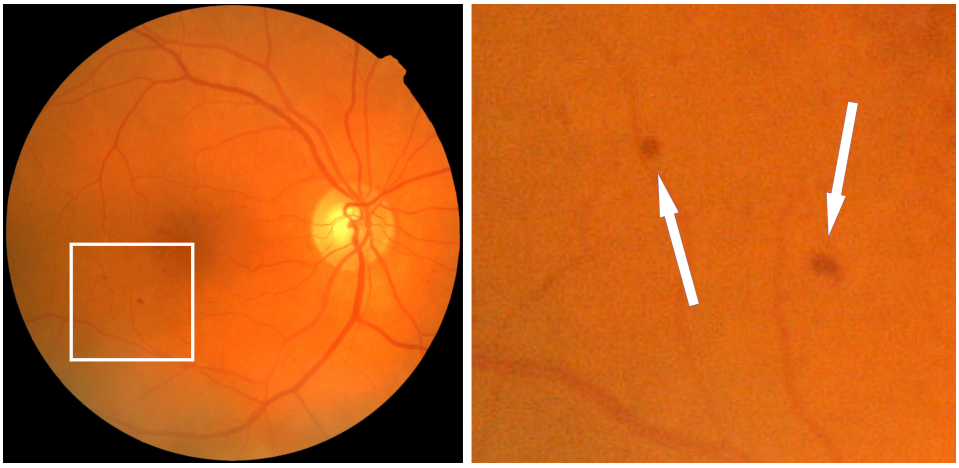


Figure 3.2. MAs in a retinal image.

seemingly simpler problem of classifying retinal images into *healthy* and *diseased* categories is not yet considered to have been solved. Automatically selecting and prioritizing cases with a higher likelihood of disease could significantly facilitate the detection of DR in a mass screening scenario because only approximately 35% of patients with diabetes mellitus have DR [34].

MAs are tiny swellings in the blood vessels (see Fig. 3.2) and the earliest clinical signs of DR, where the number of MAs is strongly correlated with its severity [41]. Consequently, the accurate detection of MAs is crucially important for recognizing DR, especially in its early stage.

Several methods have been developed to directly screen for DR based on the presence of MAs. The method proposed by Hipwell *et al.* [42] is based on the results reported by Cree [43] and it can detect MAs using red-free retinal images. After removing variation in the background intensity, small round objects are extracted as candi-

dates. Each MA candidate is then classified using intensity and size features. Fleming *et al.* [44] proposed a method that uses contrast normalization and vessel removal to improve MA detection, and they also evaluated their method for image classification. The method developed by Bhalerao *et al.* [45] is based on filtering using complex-valued circular-symmetric filters and morphological analysis of the candidate regions to reduce the false positive rate. In particular, they aimed to detect severe, sight-threatening DR. Giancardo *et al.* [46] proposed a method that discards the background areas, before calculating the Radon transform and extracting a feature vector, which is subsequently classified using principal component analysis and a nonlinear support vector machine.

The results obtained by the methods mentioned above confirm that MA detection is a reasonable approach for DR pre-screening. For further details of the performance of MA-based DR classification methods, see Section 3.4.4.

A possible approach for further increasing the accuracy of MA detection involves creating an ensemble of detectors based on different working principles and models. To demonstrate the efficiency of the proposed method, in our case study application, we considered two ensembles for the binary classification of retinal images into *healthy* or *diseased* categories based solely on the presence of MAs.

Next, we describe the members of our ensembles, the steps in the ensemble creation process, and the design choices required to implement the stochastic search method.

3.3.2 Ensemble creation method

We considered two MA detector ensembles with nine and ten members, respectively. The nine members of *Ensemble 1* were based on

traditional object detector methods [25]. This ensemble was extended to *Ensemble 2* by adding one more detector based on the fusion of two deep convolutional neural networks (DCNNs).

Member algorithms

The traditional MA detectors in our ensembles were formed as ⟨preprocessing method, candidate extractor⟩ pairs (⟨PP, CE⟩) as recommended in a previous study [2]. A ⟨PP, CE⟩ pair applied the PP to the input retinal image and the CE to its output; thus, a ⟨PP, CE⟩ pair extracted a set of MA candidates by acting as a single detector algorithm. The individual ⟨PP, CE⟩ detectors comprised the following components:

- PPs: Contrast limited adaptive histogram equalization (CLAHE) [47]; Illumination equalization (IE) [47]; Vessel removal with inpainting (VR) [48] [49]; Walter-Klein (WK) [50]; No preprocessing (NP).
- CEs: Lázár *et al.* [51]; Walter *et al.* [52]; Zhang *et al.* [53].

To extend our former ⟨PP, CE⟩ ensemble [25] with a member based on deep neural networks, we employed the method proposed by Harangi *et al.* [54], which organizes two DCNNs into a single architecture by connecting them in a shared fully connected layer in order to recognize MAs in retinal images. The advantage of this approach is that the combined architecture can be trained as a single neural network, where the training of both DCNNs is affected by the predictions of each, thereby improving the detection accuracy. The input retinal image was divided into subimages to provide the required input for the combined DCNN. An input image was labeled as *diseased* if the presence of an MA was predicted in any of the subimages at a confidence level threshold of 0.5 to 0.95, depending on its parameter.

It should be noted that MAs are dot-like lesions (especially in lower resolution retinal images), so the MA detector components of our ensembles were implemented to extract the MA centers (i.e., the coordinates of a pixel) as candidates instead of image sub-regions.

Table 3.1 summarizes the members of the two ensembles used in our study. Ensemble 1 comprised nine MA detectors $D_1, \ldots, D_9$ with the indicated $\langle$PP, CE$\rangle$ pairs (see also [25]), and Ensemble 2 included an additional DCNN member D_{10}.

Table 3.1. Members of the ensembles.

		Comp.	PP	CE	Parameter domain
Ensemble 2	Ensemble 1	D_1	NP	Lázár *et al.*	$\Pi_1 = \{1, 2, \ldots, 20\}$
		D_2	CLAHE	Lázár *et al.*	$\Pi_2 = \{1, 2, \ldots, 20\}$
		D_3	IE	Lázár *et al.*	$\Pi_3 = \{1, 2, \ldots, 20\}$
		D_4	VR	Lázár *et al.*	$\Pi_4 = \{1, 2, \ldots, 20\}$
		D_5	NP	Walter *et al.*	$\Pi_5 = \{1, 2, \ldots, 30\}$
		D_6	CLAHE	Walter *et al.*	$\Pi_6 = \{1, 2, \ldots, 30\}$
		D_7	NP	Zhang *et al.*	$\Pi_7 = \{1, 2, \ldots, 10\}$
		D_8	VR	Zhang *et al.*	$\Pi_8 = \{1, 2, \ldots, 10\}$
		D_9	WK	Zhang *et al.*	$\Pi_9 = \{1, 2, \ldots, 10\}$
		D_{10}	NP	Harangi *et al.*	$\Pi_{10} = \{0, 1, \ldots, 5\}$

The detectors listed in Table 3.1 have various numbers of parameters. However, to make the optimization problem more tractable, we considered only that parameter for each detector that had the most significant effect on the output.

In particular, the parameters $\pi_1, \ldots, \pi_4$ control thresholds for the scores assigned to the MA candidates, π_5 and π_6 control size thresholds for the diameter closing results, $\pi_7, \ldots, \pi_9$ control thresholds for the correlation map of the image used to extract candidates, and π_{10} controls the confidence threshold for MA candidates. The possible settings for each $\pi_i \in \Pi_i$ $(i = 1, \ldots, 10)$ are shown in Table 3.1. Overall,

there are $20^4 \times 30^2 \times 10^3$ and $20^4 \times 30^2 \times 10^3 \times 6$ possible different parameter settings for Ensembles 1 and 2, respectively.

Aggregation method

In order to fuse the MA candidates identified by the individual detectors $D_1^{(\pi_1)}, \ldots, D_{10}^{(\pi_{10})}$ for a given image λ via $\mathcal{D}^{(\pi)}(\lambda) = \cup_{i=1}^{10} D_i^{(\pi_i)}(\lambda)$, we need to define a confidence measure to describe the agreement of the members regarding the candidates. To this end, we first introduce a proximity relation $\cong$ to decide whether or not two candidates indicate the same MA object.

Definition 3.4. *Let c_1 and c_2 be two MA candidates. We say that c_1 and c_2 indicate the same MA object, denoted as $c_1 \cong c_2$, if their Euclidean distance is below a predefined threshold.*

Definition 3.5. *The confidence of the ensemble $conf_{\mathcal{D}^{(\pi)}}(c)$ regarding any of its candidates $c \in \mathcal{D}^{(\pi)}(\lambda)$ is defined as*

$$conf_{\mathcal{D}^{(\pi)}}(c) = |\{D_i^{(\pi)} \in \mathcal{D}^{(\pi)} : \exists c' \in D_i^{(\pi)}(\lambda) : c \cong c'\}|/|\mathcal{D}^{(\pi)}|. \quad (3.14)$$

That is, $conf_{\mathcal{D}^{(\pi)}}(c)$ for a candidate c is calculated by dividing the number of ensemble members that have a candidate c' in their respective output that indicates the same object by the total number of members.

The ensemble candidates $\mathcal{D}^{(\pi)}(\lambda)$ are classified based on the degree of confidence for the subsequent labeling of the image.

Definition 3.6. *The α–level candidates of $\mathcal{D}^{(\pi)}$ are defined as*

$$\left(\mathcal{D}^{(\pi)}(\lambda)\right)_\alpha = \{c \in \mathcal{D}^{(\pi)}(\lambda) \ : \ conf_{\mathcal{D}^{(\pi)}}(c) \geq \alpha\}, \quad (3.15)$$

where $1/|\mathcal{D}^{(\pi)}| \leq \alpha \leq 1$.

3.3.3 SA design choices

The SA-related design choices were made according to the description in Section 2.1.1. We adjusted and implemented the corresponding components of Algorithm 2 as follows.

- *Input – Initial state* π_{init}: For each member of the ensemble, a valid parameter value is randomly selected to form an initial state.

- *Input – Initial value of the control parameter* $T^{(0)}$: By using (2.1), we calculate $T^{(0)}$ as

$$T^{(0)} = -\frac{\Delta E_{max}}{\ln(\chi_0)} = -\frac{1}{\ln(0.8)} \approx 4.5, \tag{3.16}$$

  where we note the maximum possible energy difference between any two states $\Delta E_{max} = 1$ because the energy lies in the interval $[0, 1]$ in our case (see Section 3.4).

- *Line 6 – Termination criterion* OUTER-LOOP CRITERION: The algorithm stops when the temperature falls below its final value T_{final}. By using (2.2), we calculated the final temperature as

$$T_{final} = -\frac{\Delta E_{min}}{\ln(\chi_{final})} \quad \text{with} \quad \Delta E_{min} = \frac{N-1}{N}. \tag{3.17}$$

  If we set $\chi_{final} = 10^{-1000}$ and consider that in the case of a large population $\frac{N-1}{N} \approx 1$, we obtain:

$$T_{final} = -\frac{1}{\ln(10^{-1000})} \approx 0.00043. \tag{3.18}$$

- *Line 10 – Thermal equilibrium criterion* INNER-LOOP CRITERION: This criterion is omitted in our implementation. The statements in the inner loop are executed once.

- *Line 11 – Neighborhood function* GENERATE_NEIGHBOR: We define a neighborhood with a size that decreases linearly in inverse proportion to the number of search iterations. For each parameter of the ensemble, a maximum distance is determined within which a new valid parameter value is randomly selected in each iteration. This distance is the length of the range of the parameter multiplied by (1 – the ratio of the index of the current search step and the maximum number of search steps).

- *Line 15 – Acceptance function* ACCEPT: We employ the Metropolis criterion defined by (2.3), adapted to our maximization problem, because of its widespread use and attractive properties [27].

- *Line 20 – Temperature function* UPDATE_TEMPERATURE: We use the exponential temperature function defined by (2.5) to form a cooling schedule. The factor α is determined so that the search has exactly $k_{max} = 1000$ iterations:

$$\alpha = \left(\frac{T_{final}}{T^{(0)}}\right)^{\frac{1}{1000}} \approx 0.997. \tag{3.19}$$

3.4 Experimental results

In this section, we present the methods and results of our experiments. First, we describe the datasets employed, then discuss the assessment of the proposed method by performing parameter optimization of our ensembles for DR pre-screening and MA detection and provide the cor-

responding experimental results. Finally, we give some implementation details.

3.4.1 Datasets

Parameter optimization was performed for the ensembles using the publicly available dataset e-ophtha-MA [55] and the test part of the dataset provided by EyePACS for a DR grading competition held by Kaggle [56]. We will refer to the latter dataset as Kaggle EyePACS in the following. The contents of the two datasets are described as follows.

- *e-ophtha-MA*: The e-ophtha-MA dataset comprises 381 color retinal images with four different resolutions ranging from 1440 × 960 to 2544 × 1696 pixels, where 233 images depict healthy retinas (R0 class) and 148 images show various severity levels of DR (R1–R4 classes) containing a total of 1306 MAs. We used this dataset mainly because it contains precise MA ground truth data for the images.

- *Kaggle EyePACS*: The Kaggle EyePACS dataset comprises 35 126 color retinal images with various resolutions ranging from 400 × 315 to 5184 × 3456 pixels, where 25 810 images are labeled as healthy (R0), 2443 as mild DR (R1), 5292 as moderate DR (R2), 873 as severe DR (R3), and 708 as proliferative DR (R4). The images in this dataset were acquired under various imaging conditions using different models and types of cameras. Furthermore, as stated in the dataset description [56], some images are labeled incorrectly, affected by artifacts, out of focus, underexposed, or overexposed (see Fig. 3.3). According to previous studies using this dataset (e.g., see [57]) approximately 20–30% of the images are of poor quality or have incorrectly assigned labels. We used this dataset mainly because to

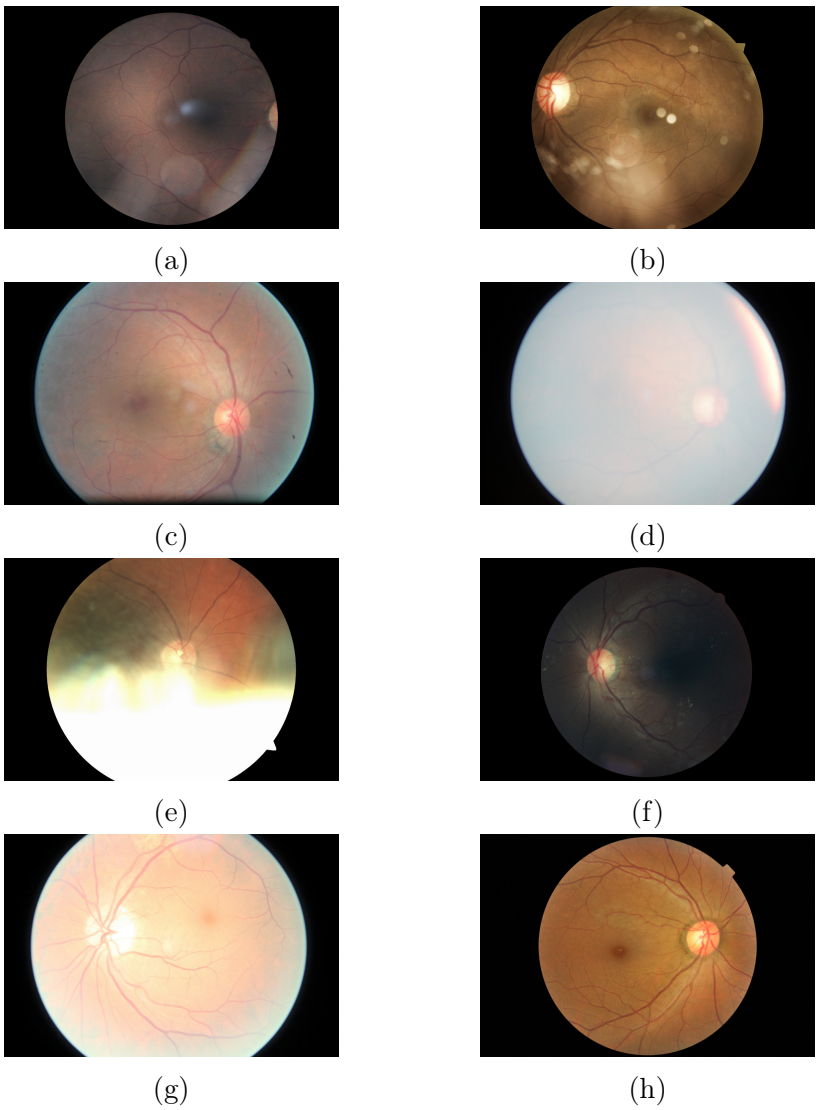


Figure 3.3. Sample images from the Kaggle EyePACS dataset showing typical artifacts and imaging errors: (a) camera artifacts, (b) lens condensation, (c) dust, (d) blur, (e) reflection, (f) underexposure, (g) overexposure, and (h) no artifacts.

the best of our knowledge, this is the largest freely available dataset that contains DR severity label ground truth data for the images.

Despite the known issues with Kaggle EyePACS, we used the images from this dataset as provided and did not perform any resource-demanding data cleaning steps (e.g., manually filtering the gradable images) because our main aim was to show that the proposed evaluation method can preserve the achievable solution quality while reducing the runtime. Clearly, due to the high proportion of poor quality or incorrectly labeled images, a lower diagnostic efficiency can be expected for Kaggle EyePACS than e-ophtha-MA using either the standard SA or the proposed method.

The contents of the datasets used in the experiments described in Sections 3.4.2 and 3.4.3 are summarized in Table 3.2 and Fig. 3.4.

Table 3.2. Contents of the datasets.

Dataset	**Subset**	**Healthy**	**Diseased**					**Total**
		R0	**R1**	**R2**	**R3**	**R4**	**R1-R4**	
e-ophtha-MA		148	-	-	-	-	233	381
	training	100	-	-	-	-	100	200
	test	48	-	-	-	-	48	96
	not used	-	-	-	-	-	85	85
Kaggle EyePACS		25810	2443	5292	873	708	9316	35126
	training	6211	1629	3528	582	472	6211	12422
	test	3105	814	1764	291	236	3105	6210
	not used	16494	-	-	-	-	-	16494

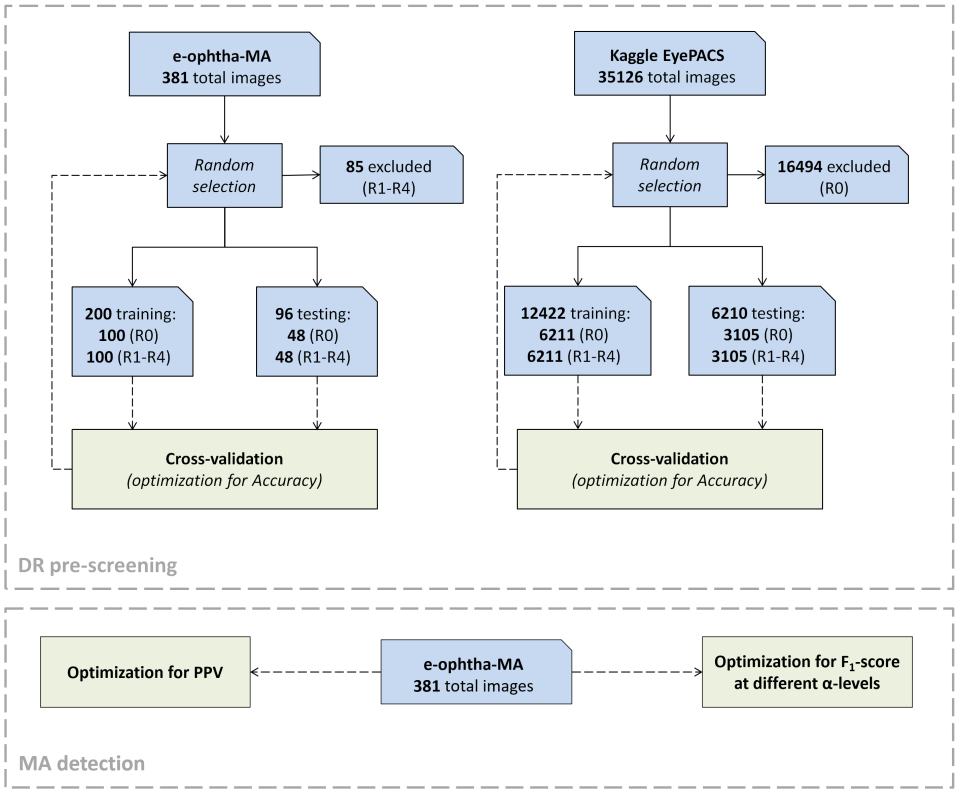


Figure 3.4. Visual overview of the datasets and the main evaluation approaches used in our experiments.

3.4.2 DR pre-screening

For DR pre-screening, the aim of the optimization process was to find the parameter setting π that maximizes the performance of the ensemble $\mathcal{D}^{(\pi)}$ in terms of the diagnostic efficiency, i.e., maximizing the proportion of correctly classified images.

The output of the ensemble was a Bernoulli distributed random variable, where $X_{\mathcal{D}^{(\pi)}} = 1$ for correct classification and $X_{\mathcal{D}^{(\pi)}} = 0$ for incorrect classification.

We considered that an image λ was classified correctly if it was annotated as positive in the ground truth and $|\left(\mathcal{D}^{(\pi)}(\lambda)\right)_\alpha| \geq 1$ (true positive), or annotated as negative and $|\left(\mathcal{D}^{(\pi)}(\lambda)\right)_\alpha| = 0$ (true negative). By contrast, λ was classified incorrectly if it was annotated as positive in the ground truth and $|\left(\mathcal{D}^{(\pi)}(\lambda)\right)_\alpha| = 0$ (false negative case), or annotated as negative and $|\left(\mathcal{D}^{(\pi)}(\lambda)\right)_\alpha| \geq 1$ (false positive case). The candidates for the ensembles were extracted at the confidence level of $\alpha = 0.5$, i.e., we used simple majority voting for this aim. The ensembles considered that an image was diseased if at least one MA was detected.

To optimize the DR pre-screening performance of the ensembles, we used the energy estimate $\widehat{E}_{\Lambda_n}$ defined in (3.2) corresponding to this implementation. It should be noted that in this case, the energy is equivalent to the accuracy (ACC) measure given as

$$ACC = \frac{\text{number of true hits}}{\text{number of all images}} = \frac{TP + TN}{TP + TN + FP + FN}, \qquad (3.20)$$

where TP, TN, FP, and FN are the numbers of true positive, true negative, false positive, and false negative hits, respectively.

Furthermore, we calculated the sensitivity (SE) and specificity (SP) measures as

$$SE = \frac{TP}{TP + FN}, \quad SP = \frac{TN}{TN + FP}. \tag{3.21}$$

For further details of ACC, SE, and SP, please refer to [58].

To evaluate the proposed method, we conducted 10-times cross-validation with repeated random subsampling of the datasets. For each round of the cross-validation process, we created new training and test subsets from the datasets described in Section 3.4.1, with a training to test ratio of approximately 2:1 (see Fig. 3.4). In the case of e-ophtha-MA, 85 randomly selected images from the R1–R4 classes in the dataset were excluded from each round in order to ensure that we had the same number of images in the R0 and R1–R4 classes. Next, 100 images were randomly selected from each of the R0 and R1–R4 classes for the training subset and the remaining 48 in each were used for testing. In the case of Kaggle EyePACS, 16494 randomly selected images from the R0 class in the dataset were excluded in each round for the same reason explained above for e-ophtha-MA. Next, 6211, 1629, 3528, 582, and 472 images were randomly selected from the R0, R1, R2, R3, and R4 classes, respectively, for the training subset and the remaining 3105, 814, 1764, 291, and 236 images were used for testing.

The optimal parameter settings obtained in each round of the cross-validation process using a training subset were evaluated using the corresponding test subset.

The main results obtained in these experiments are summarized in Tables 3.3 and 3.4. In these tables, we present the average ACC, average SE, and average SP values, as well as the average runtimes t (in seconds) and the corresponding standard deviations calculated based on the results of the 10-times cross-validation using the e-ophtha-MA

and the Kaggle EyePACS datasets, respectively. The runtimes for the test subsets are omitted because only single evaluations were needed.

Table 3.3. DR pre-screening – Results of the 10-times cross-validation using the e-ophtha-MA dataset.

		Subset	***ACC***	***SE***	***SP***	***t* (*sec*)**
Ensemble 1	SA	training	0.862 (±0.014)	0.831 (±0.027)	0.893 (±0.021)	773.6 (±100.5)
		test	0.8125 (±0.0339)	0.7625 (±0.0486)	0.8625 (±0.0529)	-
	SA-SBE	training	0.858 (±0.0127)	0.847 (±0.019)	0.869 (±0.0342)	187.3 (±53.1)
		test	0.8115 (±0.0347)	0.7833 (±0.0458)	0.8396 (±0.0618)	-
Ensemble 2	SA	training	0.8925 (±0.014)	0.883 (±0.029)	0.902 (±0.0426)	1586.5 (±189.7)
		test	0.8448 (±0.0359)	0.825 (±0.0792)	0.8647 (±0.0545)	-
	SA-SBE	training	0.896 (±0.0089)	0.889 (±0.0262)	0.903 (±0.0329)	591.4 (±151.8)
		test	0.8813 (±0.0256)	0.8833 (±0.0468)	0.8791 (±0.0445)	-

Tables 3.3 and 3.4 clearly suggest that the proposed method preserved the quality of the solution obtained using the standard SA but with significantly lower time requirements. In addition, SA-SBE exhibited stable behavior in terms of the standard deviations of ACC, SE, and SP, and also when compared to the standard SA. It should

Table 3.4. DR pre-screening – Results of the 10-times cross-validation using the Kaggle EyePACS dataset.

		Subset	***ACC***	***SE***	***SP***	***t***
Ensemble 1	SA	training	0.6516 (±0.0047)	0.5697 (±0.022)	0.7336 (±0.0243)	11936.2 (±932.9)
		test	0.6441 (±0.0125)	0.5622 (±0.0319)	0.726 (±0.0249)	-
	SA-SBE	training	0.6488 (±0.0064)	0.5643 (±0.0249)	0.7334 (±0.0299)	1685.4 (±710)
		test	0.6396 (±0.0041)	0.5556 (±0.0282)	0.7236 (±0.0314)	-
Ensemble 2	SA	training	0.6701 (±0.0068)	0.5556 (±0.0307)	0.7846 (±0.0251)	87198.2 (±9111.4)
		test	0.6649 (±0.0079)	0.5511 (±0.0250)	0.7787 (±0.0215)	-
	SA-SBE	training	0.6672 (±0.0074)	0.5476 (±0.0212)	0.7869 (±0.0216)	9611.4 (±3567.2)
		test	0.6580 (±0.006)	0.5415 (±0.0282)	0.7745 (±0.0231)	-

be noted that the average ACC was lower using Kaggle EyePACS than e-ophtha-MA because of the artifact issues discussed in Section 3.4.1. However, there were no significant differences between the average ACC values obtained with the two optimization methods. The differences in the performance of SA and SA-SBE are also highlighted in Table 3.5.

Table 3.5. Comparison of SA and SA-SBE in terms of the average solution quality and runtime based on 10-times cross-validation.

		e-ophtha-MA (training)		**Kaggle EyePACS (training)**	
		$\boldsymbol{ACC}$	$\boldsymbol{t}$ $\boldsymbol{(sec)}$	$\boldsymbol{ACC}$	$\boldsymbol{t}$ $\boldsymbol{(sec)}$
Ens. 1	SA	0.862	773.6	0.6516	11 936.2
	SA-SBE	0.858	187.3	0.6488	1685.4
	Difference	-0.004 (-0.46%)	-586.3 (-75.79%)	-0.0028 (-0.43%)	-10 250.8 (-85.88%)
Ens. 2	SA	0.8925	1586.5	0.6701	87 198.2
	SA-SBE	0.896	591.4	0.6672	9611.4
	Difference	0.0035 (0.39%)	-995.1 (-62.72%)	-0.0029 (-0.43%)	-77 586.8 (-88.98%)

We also checked the contribution of the DCNN member to the ensemble. Table 3.6 shows the individual performance of the DCNN approach together with those of the ensembles using the results obtained from the 10-times cross-validation with SA-SBE. With e-ophtha-MA, the individual performance of the DCNN was higher than that of the traditional image processing-based Ensemble 1. However, their combined performance (Ensemble 2) was better, especially considering the

more balanced *SE* and *SP* values. With Kaggle EyePACS, the DCNN component still performed better than Ensemble 1, and Ensemble 2 obtained the highest performance with an improvement in *SP*, although the performance gain was less remarkable with this dataset.

Table 3.6. Comparison of the DR pre-screening performance of the ensembles and the DCNN member.

	e-ophtha-MA (test)			**Kaggle EyePACS (test)**		
	ACC	***SE***	***SP***	***ACC***	***SE***	***SP***
Ensemble 1	0.8115	0.7833	0.8396	0.6396	0.5556	0.7236
DCNN	0.8427	0.7458	0.9396	0.6536	0.6577	0.6496
Ensemble 2	0.8813	0.8833	0.8791	0.6580	0.5415	0.7745

3.4.3 MA detection

In Section 3.4.2, we presented evaluations of our sampling-based search strategy via the optimization of our ensembles for DR pre-screening. Next, we demonstrate that the same ensembles can also be optimized using our approach for the accurate detection of MAs. We used the whole e-ophtha-MA dataset in these experiments.

The α–level candidates of an ensemble extracted using the parameter setting π for an image λ $\left(\mathcal{D}^{(\pi)}(\lambda)\right)_{\alpha}$ were compared with a set of MA centers (which were extracted from the ground truth masks provided for the image) using a method similar to that described for the fusion of MA candidates in Section 3.3.2. If the Euclidean distance between the centers of a candidate and a manually annotated MA was less than a given threshold, it was considered a true positive, otherwise a false

positive. Furthermore, each missed annotated MA was considered a false negative. The threshold was set to 5 pixels for our experiments, where this value was selected according to the average MA size in the images.

First, we optimized the parameter settings for our ensembles to maximize the mean positive predictive value $\overline{PPV}$ (see [58]) over a set of n images, i.e., the average percentage of true MAs in the output of the detector ensemble:

$$\overline{PPV} = \frac{1}{n} \sum_{i=1}^{n} \frac{TP_{\lambda_i}}{TP_{\lambda_i} + FP_{\lambda_i}}, \tag{3.22}$$

where λ_i is the i-th image, and TP_{λ_i} and FP_{λ_i} are the numbers of true positive and false positive MA candidates, respectively, in the output of the ensemble for the image λ_i.

We repeated the parameter optimization process four times with both ensembles. Table 3.7 shows the best lesion-level performance obtained with Ensemble 1 and Ensemble 2 for $\overline{PPV}$ at α–level = 0.5. Our conclusion based on these results is similar to that for the image-level results where significant reductions in the computational time were achieved with SA-SBE while the quality of solution obtained with the standard SA was preserved.

Table 3.7. MA detection performance of the ensembles using the e-ophtha-MA dataset.

	Ensemble 1		**Ensemble 2**	
	$\overline{\boldsymbol{PPV}}$	$\boldsymbol{t}$ **(sec)**	$\overline{\boldsymbol{PPV}}$	$\boldsymbol{t}$ **(sec)**
SA	0.9921	1451	0.9974	6743
SA-SBE	0.9895	172	0.9974	238

$\overline{PPV}$ is useful for optimizing our ensembles for a DR pre-screening approach based solely on the presence of MAs because a low number of false positives is a desirable characteristic of this type of system. However, $\overline{PPV}$ only considers the ratio of the number of true positives relative to the number of all positives, whereas the number of false negatives is ignored. Thus, if the ensemble finds some true positives in each image and no false positives, then $\overline{PPV}$ is 1, even if the ensemble misses numerous MAs in the images. Therefore, it would be misleading to use $\overline{PPV}$ only to assess the MA detection performance of the ensembles.

Thus, we also performed optimization for the mean F_1–score ($\overline{F_1}$) over a set of n images. $\overline{F_1}$ was considered an appropriate measure for our study because it is the average harmonic mean of PPV and SE calculated as

$$\overline{F_1} = \frac{1}{n}\sum_{i=1}^{n}\frac{2\,TP_{\lambda_i}}{2\,TP_{\lambda_i} + FP_{\lambda_i} + FN_{\lambda_i}}, \tag{3.23}$$

where the previously defined notations apply and FN_{λ_i} denotes the number of false negative MA candidates on λ_i. Fig. 3.5 shows examples of true positive, false positive, and false negative MA candidates.

Based on the optimization results obtained for $\overline{F_1}$, Fig. 3.6 shows the respective free-response receiver operating characteristic (FROC) curves [59] for Ensemble 1 and Ensemble 2, where SE is plotted against the average number of false positives per image (FPI). To measure the SE at different average FPI levels, we looped the α–level confidence value of the ensembles from 0.1 to 1 with a step size of 0.1 and repeated the optimization process accordingly. The higher performance of Ensemble 2 compared with Ensemble 1 is clearly visible in Fig. 3.6.

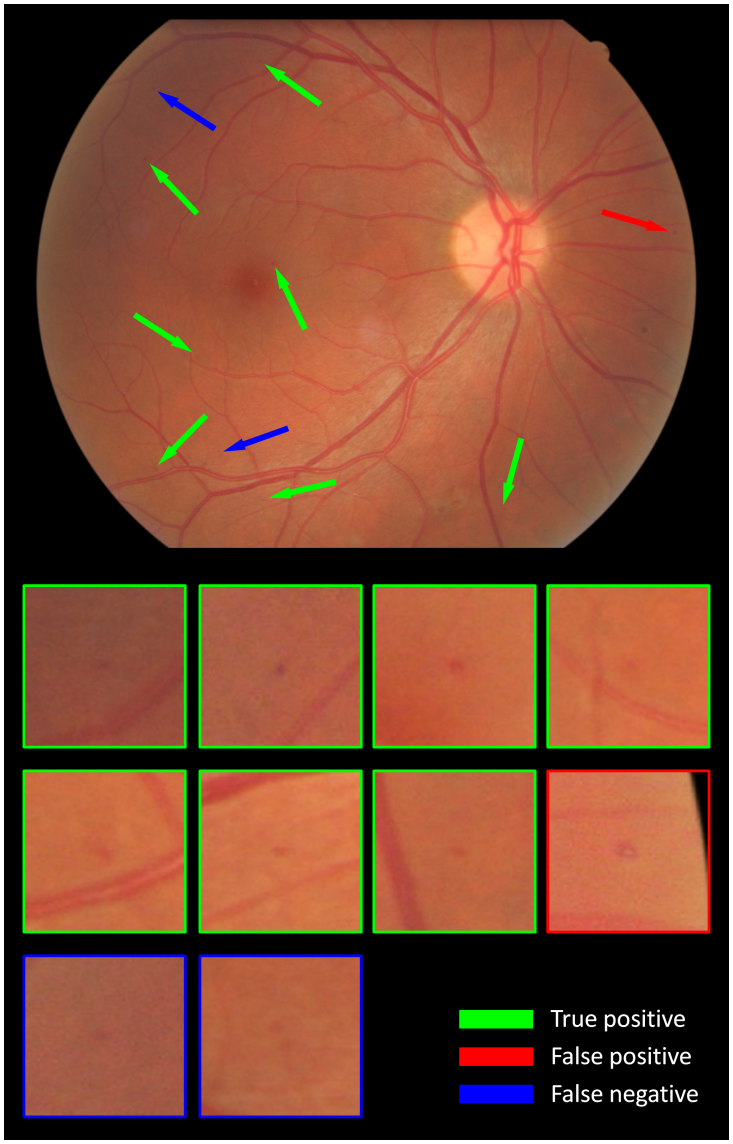


Figure 3.5. Examples of true positive, false positive, and false negative MA candidates found in an image from the e-ophtha-MA dataset by Ensemble 2.

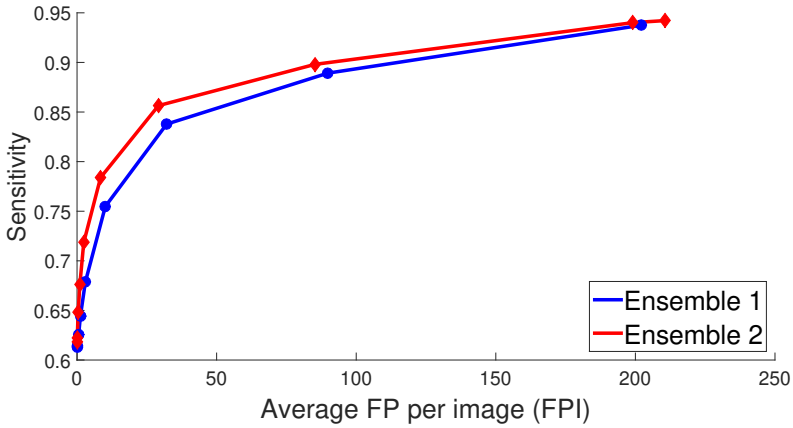


Figure 3.6. MA detection performance – FROC curves obtained for Ensemble 1 (blue) and Ensemble 2 (red).

3.4.4 DR classification at different confidence levels

In an additional experiment, we evaluated the DR classification performance of our ensembles at different confidence levels.

In this experiment, we repeated the parameter optimization process four times with both ensembles for ACC using the whole e-ophtha-MA dataset and α–level = 0.5. Using the parameter setting with the highest ACC value in the four tests, we measured ACC, SE, and SP at α–levels ranging from 0.1 to 1 with a step size of 0.1. The corresponding results are provided in Table 3.8. Furthermore, the fitted receiver operating characteristic (ROC) curves obtained for the ensembles are presented in Fig. 3.7, which again showed that Ensemble 2 performed better than Ensemble 1.

Finally, Table 3.9 gives the DR classification performance of Ensemble 2 at α–level = 0.5 and those of the methods described in Section 3.3.1. The reported performance levels are not directly comparable be-

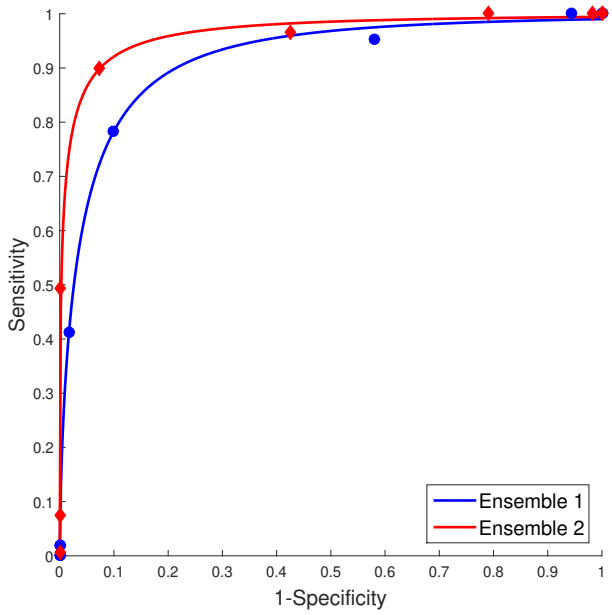


Figure 3.7. DR classification performance – ROC curves obtained for Ensemble 1 (blue) and Ensemble 2 (red) using the e-ophtha-MA dataset.

Table 3.8. DR classification performance of the ensembles at different α–levels using the e-ophtha-MA dataset.

	α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ens. 1	*SE*	1	1	1	0.9527	0.7838	0.4122	0.0203	0.0203	0	0
	SP	0	0	0.0558	0.4206	0.9013	0.9828	1	1	1	1
	ACC	0.3885	0.3885	0.4226	0.6273	0.8556	0.7612	0.6194	0.6194	0.6115	0.6115
Ens. 2	*SE*	1	1	1	0.9662	0.8986	0.4932	0.0743	0.0068	0	0
	SP	0	0.0178	0.2103	0.5751	0.9270	1	1	1	1	1
	ACC	0.3885	0.3990	0.5170	0.7270	0.9160	0.8031	0.6404	0.6141	0.6115	0.6115

cause of the different datasets and evaluation methods employed, but it can be observed that the performance of Ensemble 2 is competitive in this field.

Table 3.9. Performance of MA-based DR classification methods.

Method	**Performance**	**dataset used**
Hipwell *et al.* [42]	*SE*: 0.78, *SP*: 0.91	non-public (3783 images, 956 with DR)
Fleming *et al.* [44]	*SE*: 0.854, *SP*: 0.831	non-public (1441 images, 356 with DR)
Bhalerao *et al.* [45]	*SE*: 0.826, *SP*: 0.802	DIARETDB1 (89 images, 80 with DR)
Giancardo *et al.* [46]	*AUC*: 0.854	Messidor (1200 images, 654 with DR)
Ensemble 2	*SE*: 0.899, *SP*: 0.927 (*AUC*: 0.965)	e-ophtha-MA (381 images, 233 with DR)

3.4.5 Implementation and hardware details

SA-SBE was implemented in Java SE 8 and also used for the SA tests with sampling disabled. All the detector outputs were stored in memory during the search and the evaluation of the energy function was parallelized at the image level in order to reduce the time required to find a solution. The reported runtimes exclude the time required for loading the input files and other overheads. The results with the e-ophtha-MA dataset were acquired using a computer equipped with two 6-core AMD Opteron 2423 HE processors and 32 GB DDR2 RAM. The results with the Kaggle EyePACS dataset were acquired using two computers, where each was equipped with a 4-core Intel Xeon W-2123 processor and 64 GB DDR4 RAM.

3.5 Conclusions

In object detection applications, it is common to optimize systems using objective functions computed as an average over a dataset. Our motivation for developing the proposed method was to provide a theoretically established way to reduce the time required for optimization without compromising the quality of the achievable solution when the dataset used is large. In Section 3.2, we proposed a sampling strategy to ensure that SA exhibits the same convergence in probability using sampling-based evaluation as that using complete evaluation. Our experimental results in Section 3.4 demonstrated that SA-SBE can provide the same solution quality as SA for our parameter optimization problems. The proposed evaluation method is domain independent and easy to adapt to problems where evaluation over large datasets is required. Our method does not incorporate complex techniques for the determination of the required sample size (e.g., monitoring changes in

energy) or sample selection (e.g., finding the critical samples in classes) to accelerate the search process.

For practical problems, it is typically possible to empirically determine a fixed sampling rate for the evaluation in SA in order to obtain solutions with adequate quality and reduce the runtime. However, using the same sample size in each iteration would not necessarily provide the same solution quality as a complete evaluation. In the case of SA, according to (2.8), the standard deviation of the energy noise must approach 0 faster than the temperature to maintain the convergence in probability, i.e., the sampling rate must approach 1 faster in our case. Clearly, for any fixed sample size $n_{const} < N$, there is a temperature level $T^{(l)}$ $(0 \leq l < k_{max})$ up to n_{const} would be larger than the minimum sample size required to maintain the convergence in probability, and thus the search would be slower than possible, and after reaching $T^{(l)}$, samples of size n_{const} will be insufficient and the search convergence will deteriorate, thereby potentially decreasing the performance.

The stochastic method presented in this chapter and the corresponding DR classification results were published in [P4]. Our preliminary studies on the application of dataset sampling to accelerate parameter optimization of ensembles, which form the basis of this method, were published in [P16] and [P18]. The deep learning-based MA detector used in Ensemble 2 was proposed in [P14]. In addition, the ensemble methodology applied in this chapter for retinal image analysis was described in [P8].

Chapter 4

Optimization with Image Downscaling

4.1 Introduction

In the previous chapter, we discussed how to use dataset sampling to accelerate optimization with SA when large datasets or energy functions that are expensive to compute are used. However, sampling is not the only way to introduce controlled noise during the search. Depending on the dataset type and the application field, different approaches can be taken in order to utilize noisy evaluation to accelerate the optimization process.

In this chapter, we consider image segmentation as the application field. To accelerate the evaluation of a solution, a pyramid representation of the dataset images is used, where evaluating on lower resolution levels results in noisy determination of the energy. Naturally, the lower the resolution, the larger the noise can be, since the segmentation using lower resolution versions of the images can be less accurate. To meet the theoretical requirements, we introduce a strategy to determine the

maximum allowed noise level, and thus the lowest image resolution that can be used, in each iteration to control the search. We will show that our method successfully reduces the time requirement of the search while preserving the quality of the solution.

As a specific application, we consider an ensemble of segmentation algorithms for the extraction of bone structures from computed tomography (CT) images. The outputs of the algorithms are binary images containing the candidate bone regions, which are aggregated by majority voting.

The remainder of this chapter is organized as follows. In Section 4.2, we describe the image pyramid-based evaluation method and the strategy for selecting the appropriate scaling levels during the search to ensure that the theoretical requirements regarding the energy noise are met. The bone segmentation ensemble is described in Section 4.3: we give an overview of the member algorithms, list their adjustable parameters, and explain the aggregation strategy. Our experimental setup and results regarding the performance of the proposed evaluation method for optimizing the segmentation ensemble are presented in Section 4.4. Finally, some conclusions are drawn in Section 4.5.

4.2 SA with downscaling-based evaluation

In this section, we describe how noisy evaluation can be exploited to accelerate optimization of a segmentation ensemble using image downscaling.

4.2.1 Nearest neighbor image pyramid

To implement downscaling, a pyramid representation of the dataset images is considered. An image pyramid is a collection of images de-

rived from a single original that are successively downscaled until a termination criterion (e.g., a desired minimum resolution) is not met.

The Gaussian pyramid [60] is the most common method for creating such image pyramids, in which each level is constructed by convolving the original image with a Gaussian-like averaging filter, followed by a subsampling step. However, since the input images also have a corresponding binary ground truth, we use the nearest neighbor method to create the levels so that the sharp boundaries of the ground truth are preserved. That is, the pixel values of a level are defined to match the original pixel whose center is the nearest to the sample position.

Definition 4.1. *We refer to a collection of $L \in \mathbb{N}$ hierarchically downscaled versions of an image as an L-level image pyramid, in which the higher the scaling level l ($l \in 0, 1, \dots L-1$), the smaller the image resolution is.*

For a visual explanation of this construction, see Fig. 4.1.

4.2.2 Scaling level selection strategy

Next, we present a method to select the appropriate scaling levels of the image pyramid during the search.

Naturally, using downscaled versions of the dataset images accelerates the evaluation. Assuming that the cost of calculating the energy E is proportional to the resolution of the input images, the calculation of the energy estimate $\widehat{E}_l$ on the l-th level version of the input images (with an associated scaling factor γ_l) has $1/\gamma_l^2$ times lower cost than the calculation of E. However, using the scaling level l introduces an energy noise d_l, which is determined as

$$d_l = \widehat{E}_l - E. \tag{4.1}$$

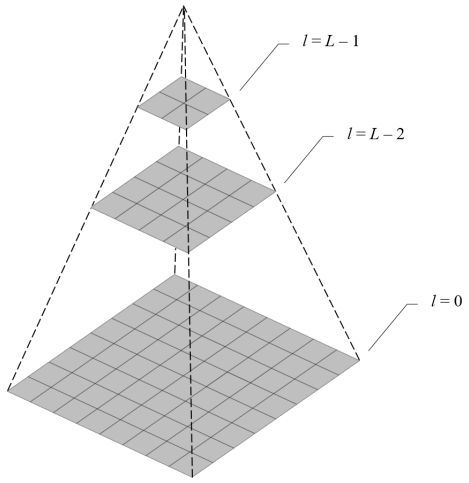


Figure 4.1. Visual explanation of the image pyramid construction.

To ensure the convergence of SA, we have to apply a strategy to select the appropriate scaling level l in each iteration k to control the noise, as described by (2.8).

First, for each k, we have to determine the maximum allowed value $s^{(k)}$ of the standard deviation $\sigma_{d_l}^{(k)}$ of the noise with respect to the temperature $T^{(k)}$. For this, we can apply *Lemma 1* from Chapter 3 as

$$s^{(k)} \gtrsim T^{(k)}(1-\epsilon)^k, \quad 0 < \epsilon \ll 1. \tag{4.2}$$

To establish the proposed evaluation method, we need to determine the standard deviation of the noise σ_{d_l} caused by downscaling for each scaling level l of the image pyramid.

It should be noted that the amount of noise can vary significantly for different energy functions when downscaled versions of the images from the dataset are used for the evaluation. In some cases, the theoretical determination of the maximum value of σ_{d_l} for a given level l

may be straightforward, while for more complex energy functions this becomes a difficult problem. Moreover, the empirical standard deviation of the noise for a level l is likely to be much lower than the theoretical maximum in the case of natural images. Therefore, even in the case when the theoretical maximum noise standard deviation can be determined, we propose to estimate σ_{d_l} for each level l of the image pyramid by measuring these values on the ground truth used for the evaluation. See Section 4.4.3 for a concrete realization of this approach.

Having σ_{d_l} measured for each level l of the image pyramid, we can determine the highest level l (i.e., the lowest resolution) where σ_{d_l} is less than or equal to the maximum allowed $\sigma_{d_l}^{(k)}$ for each temperature level $T^{(k)}$.

We refer to SA using the above described strategy as SA with Downscaling-based Evaluation (SA-DBE) in the following.

4.3 Application: bone segmentation in CT scans

In this section, we present an ensemble that performs automatic bone segmentation in CT images. We describe its member algorithms and the aggregation method used to generate the output of the ensemble based on the individual outputs of the members.

4.3.1 Member algorithms

Our automatic bone segmentation ensemble consists of five algorithms, each of which has a number of parameters. However, to gain a problem that is computationally reasonable, we selected only those parameters

for the later optimization that significantly influence the ensemble output.

Algorithm D_1: The algorithm D_1 uses distance regularized level set evolution (DRLSE) [61] with thresholding initialization and an edge-based active contour model. D_1 has the following parameters: the coefficient of the weighted area term, the width of the Dirac δ function, the coefficient of the weighted length term and a time-step parameter, which affects the coefficient of the distance regularization term. Among these parameters, the width of the δ function (π_1) is the most relevant regarding the accuracy of the output.

Algorithm D_2: This algorithm is based on the dual threshold technique described in [62] for extracting the periosteal and endosteal surfaces of the bones in two steps. We have implemented only the first step of the method for extracting the bone surface from CT images. The algorithm D_2, that applies thresholding and morphological operations, has the following parameters: the number of thresholding levels, the size of the median filter, and the parameters of the morphological structuring elements. In the case of D_2, we chose the number of thresholding levels (π_2) for the optimization.

Algorithm D_3: The algorithm D_3 uses fuzzy C-means clustering [63]. In the last step, Hounsfield-unit based thresholding of the input image is performed, and the clustering result having the least symmetric difference compared to the Hounsfield output is selected. The range for thresholding has been selected to be 500 to 900 HU [64]. D_3 has the following parameters: the number of clusters, the exponent for the fuzzy partition matrix, the iteration number and the improvement value of the objective function. Among these parameters, the number of clusters (π_3) and the exponent (π_4) have the largest influence on the output.

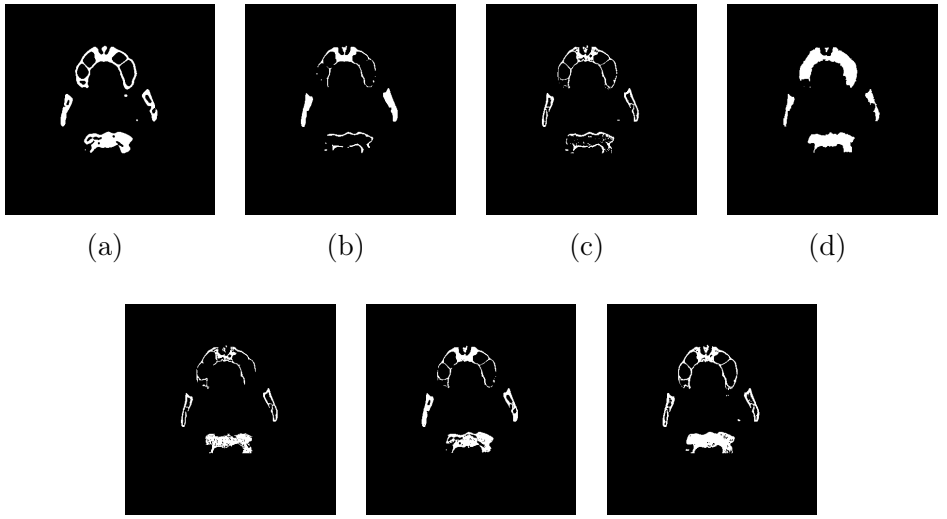
Algorithm D_4: The algorithm D_4 [65] performs histogram matching, morphological operations, and finally active contour segmentation using the method developed by Chan and Vese [66]. D_4 has the following parameters: the number of thresholding levels, the parameters of the morphological structuring elements, the weight of the smoothing term, and the number of iterations for the active contour segmentation. Among these parameters, the number of thresholding levels (π_5) has the most significant influence on the output.

Algorithm D_5: This algorithm is a variant of the region growing method [67] with multiple seed points. It compares iteratively the intensity of each unallocated neighboring pixel to the mean of the already segmented region until the difference of these values becomes larger than a threshold. Initial seed points are selected using the histogram of the input image, and the similarity threshold is automatically estimated using the variance of the input image. D_5 has two parameters: the number of initial seed points, and a correction factor of the similarity threshold, of which we chose the latter (π_6) for the optimization.

In Table 4.1, we summarize the adjustable parameters of the ensemble members.

Table 4.1. Adjustable parameters of the ensemble members.

Alg.	**Parameter description**	**Range**
D_1	width of δ function	$\pi_1 \in \{0.01, 0.21, \ldots, 2.01\}$
D_2	thresholding levels	$\pi_2 \in \{2, 3, \ldots, 5\}$
D_3	number of clusters	$\pi_3 \in \{2, 3, \ldots, 7\}$
D_3	exponent	$\pi_4 \in \{1.01, 1.21, \ldots, 3.41\}$
D_4	thresholding levels	$\pi_5 \in \{2, 3, \ldots, 5\}$
D_5	correction factor	$\pi_6 \in \{0.8, 0.9, \ldots, 1.2\}$



(a) (b) (c) (d)

(e) (f) (g)

Figure 4.2. Bone segmentation example: (a) output of D_1, (b) output of D_2, (c) output of D_3, (d) output of D_4, (e) output of D_5, (f) ensemble output, (g) ground truth.

4.3.2 Aggregation method

We chose classic majority voting as the aggregation method to obtain the output of the ensemble. That is, the pixel values of the ensemble output $\mathcal{D}^{(\pi)}(\lambda)$ for the image λ using the parameter setting π is determined as

$$\left(\mathcal{D}^{(\pi)}(\lambda)\right)_{(x,y)} = \begin{cases} 1, \text{ if } \sum_{i=1}^{M} \left(D_i^{(\pi)}(\lambda)\right)_{(x,y)} > \left[\frac{M}{2}\right], \\ 0, \text{ otherwise,} \end{cases} \tag{4.3}$$

where $M = 5$ is the number of member algorithms and (x, y) is the pixel coordinate. See Fig. 4.2 for examples of the output of the individual algorithms and the ensemble for a CT image.

4.4 Experimental results

In this section, we describe the methodology used to assess the performance of the proposed image pyramid-based noisy evaluation method, and present our quantitative results, with highlighting how the downscaling of the images affects the optimization process.

4.4.1 SA design choices

Energy function

To evaluate the proposed noisy evaluation method, we have performed parameter optimization of the ensemble presented in Section 4.3.

Our aim was to of efficiently find the parameter setting that maximizes the segmentation performance of the ensemble in terms of the intersection over union (IoU) metric [68]. Specifically, for a given level l ($l = 0, 1 \ldots, L-1$) of the image pyramid we computed the mean intersection over union $\overline{IoU}_l$, which is defined as the number of common foreground pixels of the ensemble output $\mathcal{D}^{(\pi)}(\lambda_{i,l})$ for the i-th ($i = 1, 2 \ldots, N$) image $\lambda_{i,l}$ of the dataset and the corresponding ground truth $\tau_{i,l}$ over the number of pixels in either of the two:

$$\overline{IoU}_l = \frac{1}{N} \sum_{i=1}^{N} \frac{|\mathcal{D}^{(\pi)}(\lambda_{i,l}) \cap \tau_{i,l}|}{|\mathcal{D}^{(\pi)}(\lambda_{i,l}) \cup \tau_{i,l}|}. \tag{4.4}$$

where N is the number of images in the dataset. In the case $l = 0$, we use the short notation $\overline{IoU}$.

To obtain a minimization problem, we define the energy as

$$E = 1 - \overline{IoU}, \tag{4.5}$$

and the energy estimate for the level l as

$$\widehat{E}_l = 1 - \overline{IoU}_l. \tag{4.6}$$

Cooling schedule

To implement the search, we chose the exponential cooling schedule defined by (2.5). We set the initial temperature $T^{(0)} = 1$ and the base $\alpha = 0.985$. As the stopping criterion, we chose to have a fixed number of iterations with $k_{max} = 500$.

The remaining design decisions were made according to the description in Section 2.1.1.

4.4.2 Dataset

Our dataset consists of 300 private cross-sectional CT slices in DICOM (Digital Imaging and Communications in Medicine) format taken of the head of one patient and the corresponding manually annotated ground truth masks provided by the Biomechanics Laboratory of the Faculty of General Medicine, University of Debrecen. The dataset was randomly divided into two parts: a training set with 200, and a test set with 100 images. The images have the resolution of 512×512 pixels.

The image pyramid representation of the dataset was constructed using $L = 16$ levels, with a corresponding scaling factor γ_l, defined as

$$\gamma_l = 1 - \frac{l}{L}, \quad l = 0, 1, \ldots, L-1. \tag{4.7}$$

4.4.3 Realization of the noisy evaluation

Using the setup and dataset described above, we have estimated the noise standard deviation σ_{d_l} for each level l using the ground truth of

the dataset. That is, we measured the noise originating from downscaling in the case of perfect segmentation for each level l as follows:

- Downscale the ground truth images τ_i $(i = 1, \ldots, N)$ with the scaling factor γ_l to construct the corresponding level of the image pyramids, that is, to obtain $\tau_{i,l}$.
- Upscale the images $\tau_{i,l}$ to the original size (with the scaling factor $1/\gamma_l$) to obtain $\tau_{i,l}^{'}$.
- Determine $d_{i,l}$ using (4.1) for each i. For this, compute the energy estimate $\widehat{E}_l$ using $\tau_{i,l}^{'}$ and the original ground truth image τ_i.

$$d_{i,l} = \frac{|\tau_{i,l}^{'} \cap \tau_i|}{|\tau_{i,l}^{'} \cup \tau_i|} - 1. \tag{4.8}$$

- Estimate σ_{d_l} by calculating the standard deviation of the noises $d_{i,l}$.

For comparison, we measured the noise on the aggregated output of the ensemble as well, using the parameter setting found with the proposed evaluation method (see Section 4.4.4). It can be observed in Fig. 4.3 that the standard deviation of the noise exhibits a similar trend in the case of both the ground truth and the corresponding ensemble output.

Using (4.2) and considering an exponential cooling schedule defined by (2.5), the maximum allowed value $s^{(k)}$ of $\sigma_{d_l}^{(k)}$ can be estimated as

$$s^{(k)} \approx T^{(0)}\, \alpha^k (1-\epsilon)^k \text{ with } 0 < \alpha \leq 1 \text{ and } 0 < \epsilon \ll 1. \tag{4.9}$$

Using (4.9), we can determine the required scaling level for each iteration (temperature level) k to maintain the convergence of the search while minimizing the evaluation cost by selecting the level l with the maximal corresponding standard deviation $\sigma_{d_l} \leq s^{(k)}$.

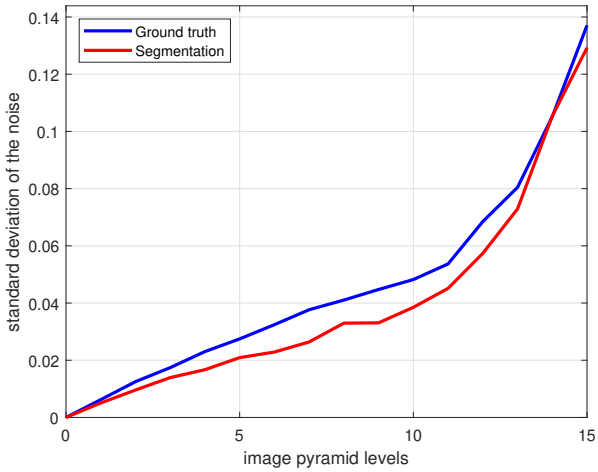


Figure 4.3. Measured standard deviation of the noise for the training set.

The maximum allowed standard deviation of the noise $\sigma_{d_l}^{(k)}$, the fitted noise σ_{d_l}, and the corresponding scaling levels l are shown in Fig. 4.4 and in Fig. 4.5, respectively.

4.4.4 Quantitative results

In Table 4.2, we give the performance of the segmentation ensemble in terms of the average sensitivity $\overline{SE}$, specificity $\overline{SP}$, Matthews correlation coefficient $\overline{MCC}$, accuracy $\overline{ACC}$, and IoU $\overline{IoU}$ using the individually optimal parameter values, and the optimal parameter values found at ensemble-level using exhaustive search, SA, and SA with the proposed noisy evaluation method, as well as the running times t in seconds required for the parameter optimization on the training set. The stochastic optimization was repeated 10 times and the run with the best $\overline{IoU}$ value is included. The running time for the individually

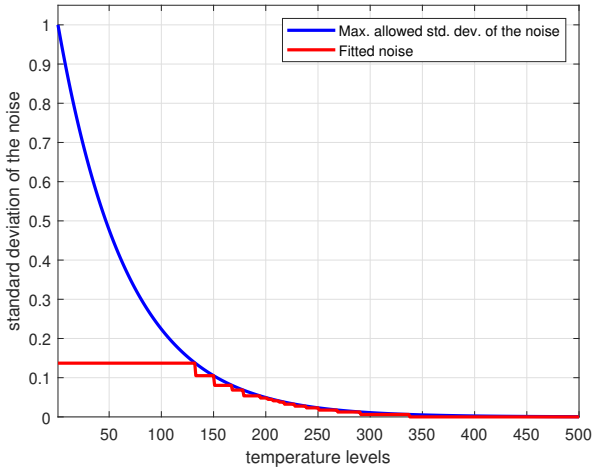


Figure 4.4. Maximal fitted standard deviation of the noise.

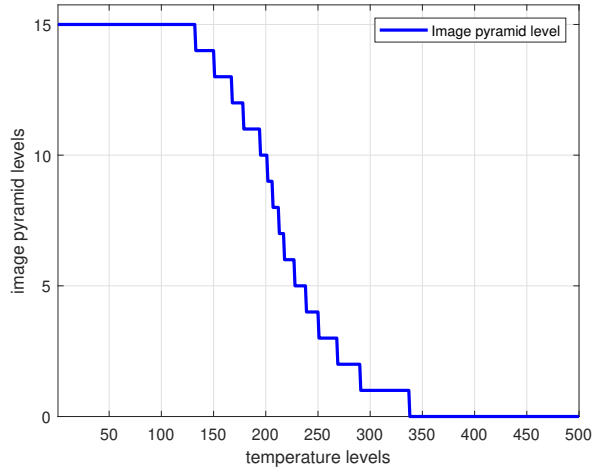


Figure 4.5. Required image size for a given temperature level during the search.

optimal parameter setup and for the test set are omitted since in these cases only evaluation was performed.

In order to assess the stability of the method, we performed 300 tests using both standard SA and SA-DBE. In Table 4.3 we include the average results of 300-300 tests. As it can be seen, the algorithm exhibits a solid behavior for the stability of the search with small differences in the average energy function values; however, significant improvement in the time requirement is achieved.

Table 4.2. Results for the dataset.

	Subset	$\overline{SE}$	$\overline{SP}$	$\overline{MCC}$	$\overline{ACC}$	$\overline{IoU}$	$t\ (sec)$
Individual evaluation	test	0.8519	0.9984	0.8827	0.9920	0.8238	-
Exhaustive search	training	0.9090	0.9970	0.9200	0.9932	0.8597	71343.7
	test	0.8588	0.9980	0.8826	0.9919	0.8235	-
SA	training	0.9085	0.9970	0.9197	0.9931	0.8591	137.3
	test	0.8583	0.9980	0.8823	0.9919	0.8230	-
SA-DBE	training	0.9021	0.9975	0.9201	0.9933	0.8589	48.9
	test	0.8509	0.9984	0.8821	0.9919	0.8226	-

Table 4.3. Performance comparison based on three hundred runs.

	$\overline{IoU}$	$t\ (sec)$
SA	0.8248 ($\pm$0.0208)	119.5 ($\pm$13.7)
SA-DBE	0.8200 ($\pm$0.0241)	39.6 ($\pm$6.8)
Difference	-0.0048 (-0.58%)	-79.9 (-66.86%)

4.4.5 Implementation and hardware details

The algorithms were implemented in Matlab. All detector outputs and ground truth images represented as image pyramids are stored in memory during the optimization process to reduce the time required to find a solution. The reported running times exclude the time required for loading the ground truth images and the algorithm output that were computed offline, generating the image pyramids, and other overhead. Results for the dataset were acquired using a computer equipped with a 4-core 8-thread Intel Xeon W-2123 processor and 16 GB DDR4 RAM.

4.5 Conclusions

In this chapter, we have proposed an image pyramid-based noisy energy function evaluation method for the local search technique SA. This method offers an alternative to dataset sampling to exploit noisy evaluation to accelerate the optimization process, especially in the case of smaller datasets.

Considering an image segmentation ensemble designed to extract bone structures from CT scans, we showed that using the proposed method it is possible to find solutions with the same quality as using the standard SA, but with a significantly reduced time requirement.

Note that the number of levels in the image pyramid should be chosen depending on the resolution of the images in the dataset and also the application domain. In general, the more levels the image pyramid has, the better the potential of noisy evaluation can be exploited. However, as the number of scaling levels increases, so does the memory requirement of the optimization process.

In the next chapter, we present a method that can fully exploit the potential of noisy evaluation by combining image downscaling using a smaller number of scaling levels with dataset sampling.

The stochastic method presented in this chapter and the corresponding bone segmentation results were published in [P11]. In addition, the lemma on the maximum allowed standard deviation of the noise in a given search iteration, which was used to develop the scaling level selection strategy, was published in [P4].

Chapter 5

Optimization with Combined Noisy Evaluation

5.1 Introduction

In Chapter 3, we proposed an evaluation method to accelerate optimization with SA based on random sampling of the dataset. This method performs evaluation on subsets of theoretically determined cardinalities of the dataset during the search and has been shown to be very efficient on large datasets. In Chapter 4, we proposed another evaluation method for SA that relies on a pyramid representation of the images in a dataset. Our experiments have shown that by using increasingly higher resolution levels of a pyramid representation of the images to evaluate solutions as the search progresses, solutions of the same quality can be found in less time than at the original input resolution. Both methods offer different approaches to noisy evaluation. They use different strategies to ensure that certain conditions regarding the resulting noise are met in order to maintain the convergence of the search.

Despite the simplicity and efficiency of the image downscaling-based evaluation compared to the standard SA, it can be shown that it does not fully exploit the potential of noisy evaluation. However, it can also be observed that image downscaling may introduce noise with a lower standard deviation than dataset sampling with the same cost gain, depending on the problem. Based on these observations, here we investigate the possible combination of these methods.

That is, in this chapter, we propose an evaluation method for SA that computes the energy using a combination of dataset sampling and image downscaling to further accelerate optimization on image datasets. For this aim, we give a strategy that is capable to determine the appropriate scaling level and sample size in each search step by adapting convergence results for noisy evaluation in SA. The proposed method is primarily intended for the optimization of image segmentation algorithms. To demonstrate the efficiency of the combined evaluation method, we consider the optimization of a system that comprises an ensemble of conventional image processing algorithms and a post-processing step to segment the lungs in computed tomography (CT) scans. Through this, we show that the proposed method further reduces the cost of the search while preserving solution quality.

The rest of this chapter is organized as follows. In Section 5.2, we describe the proposed evaluation method. Then, we present our case study application in Section 5.3. The results of our experiments are given in Section 5.4. Detailed results are provided on the efficiency of the evaluation method for the parameter optimization of the segmentation ensemble, also compared to the two previously described methods. We also give the performance of the lung segmentation method. Finally, some conclusions are drawn in Section 5.5.

5.2 SA with combined noisy evaluation

In this section, we propose a novel evaluation method for SA that combines image downscaling and dataset sampling to reduce the time required for optimization over image datasets. To maintain the convergence of SA in probability, we define a strategy that determines the appropriate scaling level and sample size in each search step by adapting convergence results for noisy evaluation in SA.

5.2.1 Combining dataset sampling and image downscaling

As in the case of the evaluation methods presented in the previous chapters, we have to control the standard deviation of the energy noise during the search according to (2.8) to ensure convergence of SA. That is, for each iteration k, we need to determine the maximum allowed value $s^{(k)}$ for the standard deviation $\sigma_d^{(k)}$ of the noise with respect to the temperature $T^{(k)}$. For this, we can apply *Lemma 1* as

$$s^{(k)} \gtrsim T^{(k)}(1-\epsilon)^k, \quad 0 < \epsilon \ll 1. \tag{5.1}$$

When combining dataset sampling and image downscaling, we have to deal with a noise that is the sum of noises originating from the dataset sampling d_n and the image downscaling d_l. Since d_n and d_l are uncorrelated, the standard deviation of the sum of these noises $\sigma_{(d_n+d_l)}$ can be calculated as the square root of the sum of their variances, that is, as

$$\sigma_{(d_n+d_l)} = \sqrt{\sigma_{d_n}^2 + \sigma_{d_l}^2}. \tag{5.2}$$

Furthermore, to maintain the convergence of the search

$$\sigma^{(k)}_{(d_n+d_l)} \leq s^{(k)} \tag{5.3}$$

must hold in each iteration k.

Our aim is to minimize the total cost of the search, e.g., in terms of computation time. For this, we need to find that scaling level l and sample size n whose combination will minimize the cost $C^{(k)}_{l,n}$ in each iteration k while ensuring (5.3). To do this, we use the method described next.

For each scaling level l whose corresponding noise standard deviation $\sigma_{d_l} \leq s^{(k)}$, we estimate the minimum required sample size n by adapting (3.10) as

$$n \approx \frac{N\sigma^2_{max}}{(N-1)\sqrt{{s^{(k)}}^2 - \sigma^2_{d_l}} + \sigma^2_{max}}, \tag{5.4}$$

and calculate the corresponding cost $C^{(k)}_{l,n} = nc_l$. Finally, we choose the combination of l and n for which $C^{(k)}_{l,n}$ is minimal.

Note that this method ensures that the scaling level l decreases monotonically during the search if the corresponding noise standard deviation d_l decreases monotonically, which is a natural assumption. Thus, in this case, the method described above can be accelerated by not calculating the cost for levels higher than the one used in the previous iteration.

Hereafter, we refer to SA using the above strategy to compute energy estimates during the search as SA with Combined Noisy Evaluation (SA-CNE).

5.2.2 Example

As a numerical demonstration of the total cost of the method described above, let us consider a problem with a maximum standard deviation of the evaluation metric $\sigma_{max} = 0.5$ and a training set size $N = 200$. For the image pyramid, let us use $L = 5$ levels with the corresponding noise standard deviations $\sigma_{d_{l=0}} = 0.0$, $\sigma_{d_{l=1}} = 0.02$, $\sigma_{d_{l=2}} = 0.05$, $\sigma_{d_{l=3}} = 0.09$ and $\sigma_{d_{l=4}} = 0.15$ and the corresponding evaluation costs $c_l = \frac{1}{2^{2l}}$ ($l \in \{1, 2, \ldots, L-1\}$). Furthermore, let us use an exponential cooling schedule in SA with

$$T^{(k)} = T^{(0)}\alpha^k, \tag{5.5}$$

where $T^{(0)} = 5$, $\alpha = 0.99$, and $0 \leq k < 1000$.

For this setup, the total theoretical cost of optimization is 200 000 when using standard SA, 101 905 when using dataset sampling, 95 719.5 when using image downscaling, and 90 902.5 when combining dataset sampling and image downscaling. The change in cost during optimization for both methods are shown in Fig. 5.1.

5.3 Application: lung segmentation in CT scans

CT is a widely used imaging modality for computer-aided diagnosis systems aimed at detecting and characterizing various lung abnormalities. Lung segmentation is a crucial step in these systems and can significantly affect their performance.

In the last two decades, several methods for lung segmentation have been proposed. Conventional methods rely on techniques such as thresholding [69], region growing [70], active contours [71], mathematical morphology [72], and cluster analysis [73]; however, deep learning

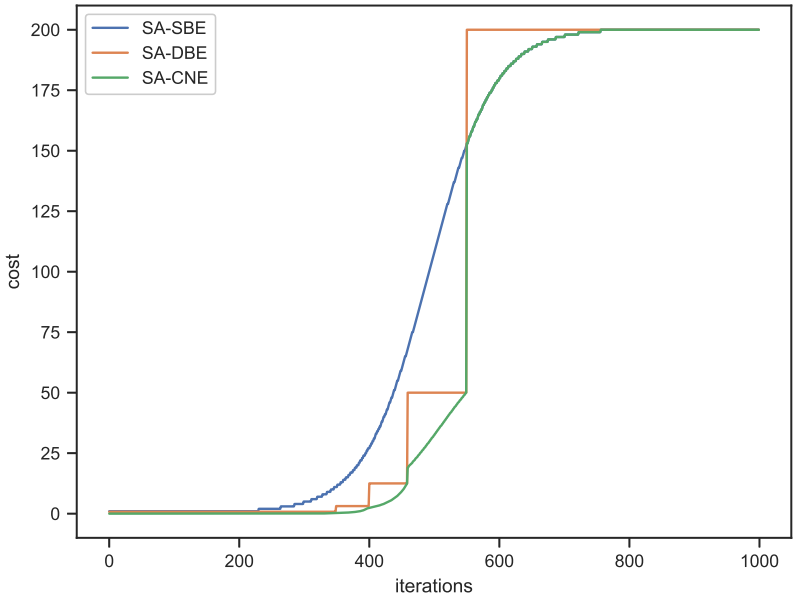


Figure 5.1. Changes of the cost during the search.

approaches, in particular convolutional neural networks [74] and generative adversarial networks [75], have recently gained popularity in this field too. While deep learning methods achieve state-of-the-art accuracy, a drawback of these methods is that they require substantial amounts of annotated training data to achieve the desired accuracy.

Another approach to improve accuracy is to create an ensemble of segmentation methods by merging their output using an aggregation rule [10]. In the rest of this section, we present our case study application, which uses an ensemble of conventional segmentation methods and a post-processing step to segment the lungs.

5.3.1 Segmentation ensemble

Next, we describe the members of the lung segmentation ensemble and the aggregation method used to combine their outputs.

Member algorithms

Our lung segmentation ensemble consists of three simple, conventional image processing algorithms, each of which works on the slices of the CT scans. These methods have been developed to have different operating principles to increase the diversity of the ensemble, i.e., the independence of the outputs of the members, and thus to reduce the segmentation error.

Algorithm D_1: This algorithm is based on connected component analysis. First, the input image is thresholded to estimate an initial lung mask, then the connected components in the resulting binary image are labeled. Those components that are touching the border of the image are removed, and the two largest remaining components are kept as lung candidates. After this, morphological erosion is performed to separate the lung nodules attached to the blood vessels, and morphological closing is used to retain nodules attached to the lung wall. Finally, the holes inside the binary mask of the lungs are filled. D_1 has the following parameters: the threshold to gain the initial lung mask π_1, the radius of the disk structuring element (SE) for erosion π_2, and the radius of the disk SE for closing π_3.

Algorithm D_2: This algorithm is based on k-means clustering. First, the image is standardized to have pixel values with mean 0 and standard deviation 1. After this, k-means clustering is used to partition the image into $k = 2$ clusters, representing lungs/air and tissues/bones. The resulting binary image is then refined using morphological erosion and a subsequent dilation using a larger SE in order to avoid missing

lung pixels. Then, the air outside the body of the subject and those objects that are too small to be lungs (less than 1% of the image area) are removed. Finally, morphological closing is performed to fill small holes. D_2 has the following parameters: the radius of the disk SEs for erosion π_4, for dilation π_5, and for closing π_6.

Algorithm D_3: This algorithm is based on contour detection. First, the intensity values of the input image are clipped to an interval that roughly represents lungs/air. That is, intensities below the minimum and above the maximum of the interval are set to these values, respectively. Then, the image is binarized, and its contours are detected using the "marching squares" method for a given level. After this, non-closed contours are removed. Using a minimum area (the square root of the image area) and a maximum area (the quarter of the image area) unwanted small contours and the contour of the body is removed. The remaining contours are assumed to correspond to the lungs. Finally, a lung mask is created converting the list of the remaining contours to a binary image. D_3 has the following parameters: the minimum π_7 and maximum π_8 values of the interval for clipping, and the level π_9 for contour detection.

In Table 5.1, we summarize the adjustable parameters of the ensemble members. Overall, there are $3^8 \times 5 = 32\,805$ possible different parameter settings for the ensemble.

Aggregation method

To obtain the output of the ensemble, we use pixel-level majority voting to aggregate the individual member outputs. That is, the pixel values of the ensemble output $\mathcal{D}^{(\pi)}(\lambda)$ for the image λ using the parameter

Table 5.1. Adjustable parameters of the ensemble members.

Alg.	Parameter description	Range
D_1	initial mask threshold	$\pi_1 \in \{-450, -400, -350\}$
D_1	erosion SE radius	$\pi_2 \in \{1, 2, 3\}$
D_1	closing SE radius	$\pi_3 \in \{8, 9, 10, 11, 12\}$
D_2	erosion SE radius	$\pi_4 \in \{1, 2, 3\}$
D_2	dilation SE radius	$\pi_5 \in \{3, 4, 5\}$
D_2	closing SE radius	$\pi_6 \in \{1, 2, 3\}$
D_3	interval minimum	$\pi_7 \in \{-1050, -1000, -950\}$
D_3	interval maximum	$\pi_8 \in \{-450, -400, -350\}$
D_3	contour level	$\pi_9 \in \{0.75, 0.85, 0.95\}$

setting π is determined as

$$\left(\mathcal{D}^{(\pi)}(\lambda)\right)_{(x,y)} = \begin{cases} 1, \text{ if } \sum_{i=1}^{M} \left(D_i^{(\pi)}(\lambda)\right)_{(x,y)} > \left[\frac{M}{2}\right], \\ 0, \text{ otherwise,} \end{cases} \tag{5.6}$$

where $M = 3$ is the number of member algorithms and (x, y) is the pixel coordinate.

5.3.2 Post-processing: removing air pockets

The previously presented ensemble performs lung segmentation on a CT slice level. However, after aggregating the segmentation outputs of the members and building a new volume from these consensus slices, the result can be improved by removing the small air pockets and retaining only those that are almost certainly lung.

For this, we use the following simple method: first, the connected components of the volume are labeled, and the voxel count for each unique label is calculated. In addition, the background is removed in

this step. Then, if there are at least two connected components left, we check whether the largest component has at least twice the voxel count of the second largest one. If it holds, the largest component is retained; that is, we assume that the lungs are not separated from the trachea. If this condition does not hold, we assume that the lungs and trachea have been separated, and we keep the two largest components. If only one label remains after removing the background, that component is considered as lung.

5.4 Experimental results

In this section we present the methods and results of our experiments. We start with a description of the dataset used, followed by a discussion of the methodology applied to evaluate the proposed optimization methods. We then present the results of our experiments. Finally, we provide the implementation details.

5.4.1 Dataset

Parameter optimization of the ensemble was performed using the publicly available *COVID-19 CT Lung and Infection Segmentation Dataset* [76]. This dataset consists of 20 thoracic CT scans and corresponding annotations, both stored in NIfTI (Neuroimaging Informatics Technology Initiative) format. The CT scans were provided by the Coronacases Initiative [77] and Radiopaedia [78]. The left and right lungs and signs of COVID-19 infection were labeled by two radiologists and reviewed by a third experienced radiologist. From this dataset, the 10 CT scans provided by the Coronacases Initiative were used for the experiments. The resolution of these CT scans ranges from $512\times512\times200$ to $512 \times 512 \times 301$ pixels, and they contain a total of 2581 slices.

In our experiments, we used 10 training sets, each created from the dataset described above as follows: first, we randomly selected 3 of the 10 CT scans, then we randomly selected 100 slices from each of these scans. In this way, we obtained training sets of 300 slices each. For each training set, the remaining 7 CT scans were used as the corresponding test set.

5.4.2 Evaluation methodology

We performed parameter optimization of the ensemble presented in Section 5.3 to assess the efficiency of the proposed stochastic search methods.

For the evaluation, we considered each CT slice as a separate image λ. The ensemble output $\mathcal{D}^{(\pi)}(\lambda)$ for the image λ using the parameter setting π was compared with the corresponding binary ground truth mask τ. That is, a pixel was considered a true positive if it was correctly classified as foreground; otherwise, it was considered a false positive. In addition, an unrecognized foreground pixel was treated as a false negative, while a correctly classified background pixel was treated as a true negative. For the i-th image $\lambda_{i,l}$ scaled to the l-th image pyramid level, we denote the number of true positives as $TP_{\lambda_{i,l}}$, the number of false positive as $FP_{\lambda_{i,l}}$, the number of false negatives as $FN_{\lambda_{i,l}}$, and the the numbers of true negatives as $TN_{\lambda_{i,l}}$. If $l = 0$, we use the short notations TP_{λ_i}, TN_{λ_i}, FP_{λ_i}, and FN_{λ_i}, respectively.

The goal of the optimization was to find the parameter setting that maximizes the segmentation performance of the ensemble in terms of the mean F_1-score. Specifically, for a given level l of the image pyramid and for a set of n images, we computed the mean F_1-score $\overline{F_1}_{l,n}$ for each

$l = 0, 1, \ldots, L-1$ as follows:

$$\overline{F_1}_{l,n} = \frac{1}{n} \sum_{i=1}^{n} \frac{2TP_{\lambda_{i,l}}}{2TP_{\lambda_{i,l}} + FP_{\lambda_{i,l}} + FN_{\lambda_{i,l}}}, \tag{5.7}$$

where $\lambda_{i,l}$ is the i-th image downscaled to the image pyramid level l, L is the number of levels in the image pyramid, and n is the number of images used. In the case $l = 0$ and $n = N$, we use the short notation $\overline{F_1}$.

Furthermore, for a solution we calculated the mean Matthews correlation coefficient $\overline{MCC}$, the mean accuracy $\overline{ACC}$, the mean sensitivity $\overline{SE}$ and mean specificity $\overline{SP}$ measures as

$$\overline{MCC} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_{\lambda_i} TN_{\lambda_i} - FP_{\lambda_i} FN_{\lambda_i}}{((TP_{\lambda_i} + FP_{\lambda_i})(TP_{\lambda_i} + FN_{\lambda_i})(TN_{\lambda_i} + FP_{\lambda_i})(TN_{\lambda_i} + FN_{\lambda_i}))^{\frac{1}{2}}}, \tag{5.8}$$

$$\overline{ACC} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_{\lambda_i} + TN_{\lambda_i}}{TP_{\lambda_i} + TN_{\lambda_i} + FP_{\lambda_i} + FN_{\lambda_i}}, \tag{5.9}$$

$$\overline{SE} = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_{\lambda_i}}{TP_{\lambda_i} + FN_{\lambda_i}}, \tag{5.10}$$

$$\overline{SP} = \frac{1}{N} \sum_{i=1}^{N} \frac{TN_{\lambda_i}}{TN_{\lambda_i} + FP_{\lambda_i}}. \tag{5.11}$$

5.4.3 SA design choices

A number of design decisions need to be made in order to implement the proposed stochastic search method.

Energy function

To obtain a minimization problem, we defined the energy function as

$$E = 1 - \overline{F_1}, \tag{5.12}$$

and the energy function estimate for the level l ($l = 1, 2, \ldots, L-1$) and a set of n images as

$$\widehat{E}_{l,n} = 1 - \overline{F_1}_{l,n}. \tag{5.13}$$

Furthermore, the number of image pyramid levels was set to $L = 5$.

Cooling schedule

For the implementation, we chose an exponential cooling schedule, using the temperature function defined in (2.5). For this, we set the initial temperature $T^{(0)} = 1$ and $\alpha = 0.985$. For the termination criterion, we chose a fixed number of iterations with $0 \leq k < 500$.

The remaining design decisions were made according to the description in Section 2.1.1.

5.4.4 Estimation of the noise caused by downscaling

To estimate the noise caused by evaluating the segmentation performance of the ensemble using downscaled versions of the dataset images, we measured the noise using the corresponding ground truth masks provided with the dataset (i.e., the noise in the case of perfect segmentation) for each level $l = 0, 1, \ldots, L-1$ as follows.

First, for a level l and for each ground truth mask τ_i ($i = 1, \ldots, N$), we generated the images $\tau_{i,l}$ by scaling them down by the factor $\gamma_l =$

$1/2^l$. These images were then scaled up to their original size to obtain $\tau'_{i,l}$. Then, for each ground truth mask τ_i, we calculated the noise $d_{i,l}$ as the difference of the energy function and its estimate as

$$d_{i,l} = \frac{2TP_{\tau'_{i,l}}}{2TP_{\tau'_{i,l}} + FP_{\tau'_{i,l}} + FN_{\tau'_{i,l}}} - 1. \tag{5.14}$$

Finally, we estimated σ_{d_l} by calculating the standard deviation of the noises $d_{i,l}$.

5.4.5 Optimization results

The main results obtained in our experiments regarding the efficiency of the proposed optimization method are summarized in Table 5.2. Here we present the average $\overline{F_1}$, $\overline{MCC}$, $\overline{ACC}$, $\overline{SE}$, and $\overline{SP}$ values, as well as the average runtimes t (in seconds) and the corresponding standard deviations calculated based on the results obtained using the 10 training sets. For each training set, the optimization was repeated three times and the run with the best energy function value was selected. The results for the solutions found with an exhaustive search (i.e., the achievable maximum performance) is also included for reference.

Table 5.2 shows that both the sampling-based, the downscaling-based, and the combined methods maintained the solution quality of the standard SA, but required significantly less time (-32.54%, -33.58%, and -42.01%, respectively). That is, the $\overline{F_1}$ values obtained with these three optimization methods were only marginally different from those obtained with the standard SA. It is also shown that the combined method could further speed up the optimization process reasonably without affecting the quality of the solution. Moreover, all methods

Table 5.2. Results of the parameter optimization using the 10 training sets.

Method	$\overline{F_1}$	$\overline{MCC}$	$\overline{ACC}$	$\overline{SE}$	$\overline{SP}$	t (sec)
Exhaustive search	0.7956 (±0.0377)	0.7737 (±0.0356)	0.9927 (±0.0015)	0.9367 (±0.0145)	0.9960 (±0.0005)	4465.8 (±25.3)
SA	0.7954 (±0.0377)	0.7735 (±0.0356)	0.9927 (±0.0015)	0.9378 (±0.0145)	0.9959 (±0.0006)	67.6 (±0.4)
SA-SBE	0.7953 (±0.0377)	0.7735 (±0.0356)	0.9927 (±0.0015)	0.9369 (±0.0148)	0.9960 (±0.0005)	45.6 (±0.5)
SA-DBE	0.7954 (±0.0378)	0.7735 (±0.0356)	0.9926 (±0.0015)	0.9363 (±0.0148)	0.9960 (±0.0005)	44.9 (±0.3)
SA-CNE	0.7955 (±0.0378)	0.7736 (±0.0356)	0.9927 (±0.0015)	0.9370 (±0.0143)	0.9960 (±0.0006)	39.2 (±0.3)

showed stable behavior in terms of the standard deviations of the evaluation measures.

5.4.6 Lung segmentation results

For each parameter setting obtained using the training sets, we used the corresponding test sets for evaluation as follows. First, we evaluated the performance of the lung segmentation ensemble without the air pocket removal (post-processing) step. That is, we processed all slices of the CT scans in the test sets using the segmentation ensemble to obtain the mean metrics for each test set. Then, these values were averaged over the 10 test sets. The corresponding results can be found in Table 5.3.

Next, we evaluated the lung segmentation performance using the volume-level air pocket removal method to improve the final segmentation performance. To accomplish this, we generated a new volume

Table 5.3. Lung segmentation performance without post-processing.

Method	$\overline{\boldsymbol{F_1}}$	$\overline{\boldsymbol{MCC}}$	$\overline{\boldsymbol{ACC}}$	$\overline{\boldsymbol{SE}}$	$\overline{\boldsymbol{SP}}$
Exhaustive search	0.7742 (±0.0159)	0.7593 (±0.0177)	0.9919 (±0.0007)	0.9314 (±0.0096)	0.9957 (±0.0004)
SA	0.7745 (±0.0157)	0.7596 (±0.0175)	0.9920 (±0.0007)	0.9322 (±0.0091)	0.9957 (±0.0003)
SA-SBE	0.7744 (±0.0157)	0.7595 (±0.0174)	0.9920 (±0.0007)	0.9316 (±0.0090)	0.9957 (±0.0003)
SA-DBE	0.7741 (±0.0158)	0.7592 (±0.0177)	0.9919 (±0.0007)	0.9309 (±0.0104)	0.9958 (±0.0004)
SA-CNE	0.7744 (±0.0158)	0.7595 (±0.0176)	0.9920 (±0.0007)	0.9318 (±0.0095)	0.9957 (±0.0003)

Table 5.4. Lung segmentation performance with post-processing.

Method	$\overline{\boldsymbol{F_1}}$	$\overline{\boldsymbol{MCC}}$	$\overline{\boldsymbol{ACC}}$	$\overline{\boldsymbol{SE}}$	$\overline{\boldsymbol{SP}}$
Exhaustive search	0.9393 (±0.0066)	0.7753 (±0.0178)	0.9939 (±0.0006)	0.9311 (±0.0097)	0.9978 (±0.0004)
SA	0.9397 (±0.0068)	0.7757 (±0.0175)	0.9939 (±0.0006)	0.9319 (±0.0092)	0.9977 (±0.0003)
SA-SBE	0.9395 (±0.0068)	0.7754 (±0.0175)	0.9939 (±0.0006)	0.9314 (±0.0091)	0.9978 (±0.0003)
SA-DBE	0.9391 (±0.0071)	0.7750 (±0.0180)	0.9939 (±0.0006)	0.9306 (±0.0104)	0.9978 (±0.0003)
SA-CNE	0.9395 (±0.0067)	0.7755 (±0.0175)	0.9939 (±0.0006)	0.9315 (±0.0095)	0.9977 (±0.0003)
U-Net (R-231) [74]	0.9651 (±0.0023)	0.8015 (±0.0155)	0.9968 (±0.0002)	0.9693 (±0.0021)	0.9979 (±0.0002)

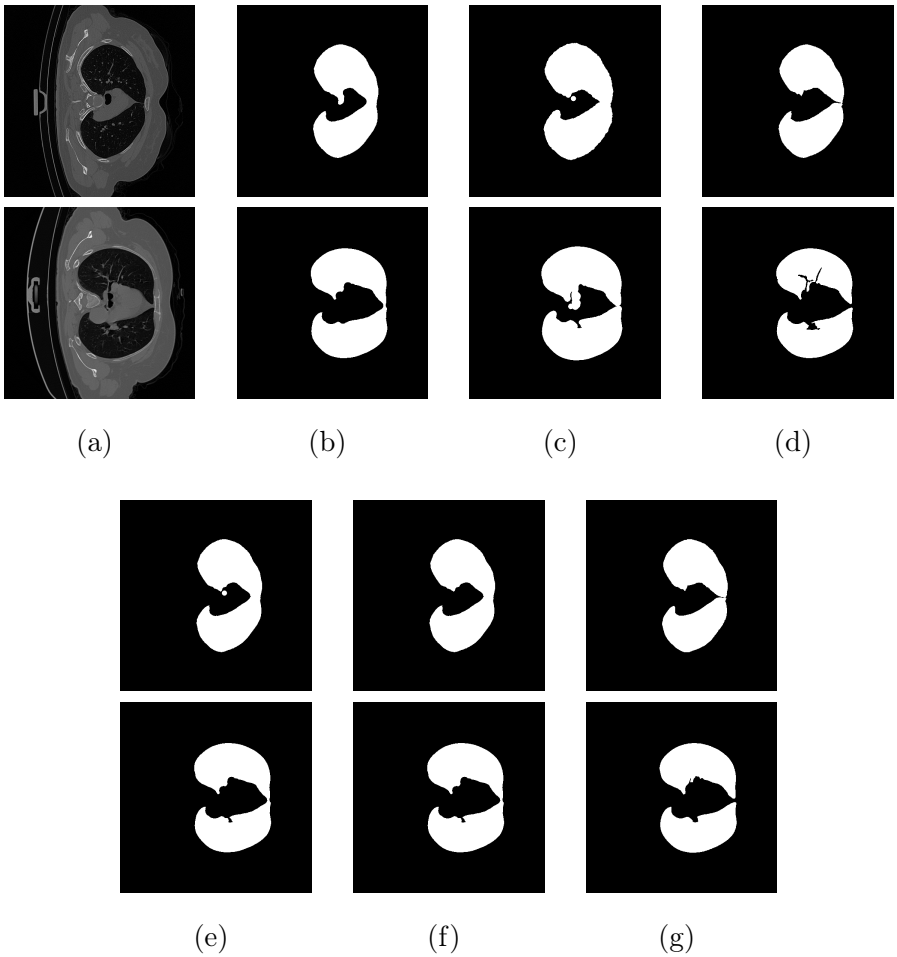


Figure 5.2. Lung segmentation examples: (a) input slices, (b) outputs of D_1, (c) outputs of D_2, (d) outputs of D_3, (e) ensemble outputs, (f) post-processed ensemble outputs, (g) ground truth.

from the output of the segmentation ensemble for a given CT scan and applied the air pocket removal method to it. For comparison, the CT scans of the test sets were also processed using the state-of-the-art deep learning method U-Net trained on a dataset of 231 CT scans [74]. In both cases, we calculated the mean metrics at the slice level. See Table 5.4 for the corresponding results. It can be observed that the post-processing step significantly improves the $\overline{F_1}$ and $\overline{MCC}$ values. It is also shown that despite the simplicity of the ensemble members, the segmentation performance is not far behind the performance of a deep learning method. For some examples of the output of the ensemble and its members, see Fig. 5.2.

5.4.7 Implementation and hardware details

All segmentation algorithms were implemented in Python and all optimization algorithms were implemented in Matlab. The outputs of the algorithms were computed offline and stored in memory during the optimization process along with the corresponding ground truth to reduce the time required to find a solution. The reported runtimes do not include the time required to load the ground truth and the outputs of the algorithms, nor any other overheads. Results for the dataset were obtained using a computer equipped with a 4-core 8-thread AMD Ryzen 3 3100 processor and 32 GB of DDR4 RAM.

5.5 Conclusions

In this chapter, we proposed an SA-based stochastic search method that combines dataset sampling and image downscaling to implement noisy evaluation of the energy. By optimizing the parameters of a lung segmentation ensemble, we have shown that the proposed search

method is capable of finding solutions with the same quality as the standard SA, but with a significantly lower time requirement.

It should be noted, that considering a given problem, the combined evaluation method reverts to the sampling-based one above a certain dataset size N_p. As a result, the proposed search method using the combined strategy is never less efficient than using the sampling-based one in terms of evaluation cost, and it is always more efficient than the sampling-based strategy when the dataset size is smaller than N_p.

Considering real-world implementations, using the combined strategy instead of the sampling-based one for datasets larger than N_p implies a negligible computational overhead during the search. However, in this case, generating and storing the pyramid representations of the dataset images is unnecessary. For this reason, it is beneficial to determine which method is more appropriate for a given problem by calculating the total cost of the two methods for the size of the dataset.

The stochastic method presented in this chapter and the corresponding lung segmentation results were published in [P1]. The stochastic optimization methods based on dataset sampling and image downscaling, as well as the related theoretical results, on which the presented method relies, were published in [P4] and [P11], respectively.

References

[1] D. West, P. Mangiameli, R. Rampal, and V. West, "Ensemble strategies for a medical diagnostic decision support system: A breast cancer diagnosis application," *European Journal of Operational Research*, vol. 162, no. 2, pp. 532–551, 2005.

[2] B. Antal and A. Hajdu, "An ensemble-based system for microaneurysm detection and diabetic retinopathy grading," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1720–1726, 2012.

[3] A. Teramoto, H. Fujita, O. Yamamuro, and T. Tamaki, "Automated detection of pulmonary nodules in PET/CT images: Ensemble false-positive reduction using a convolutional neural network technique," *Medical Physics*, vol. 43, no. 6, pp. 2821–2827, 2016.

[4] B. Harangi, "Skin lesion classification with ensembles of deep convolutional neural networks," *Journal of Biomedical Informatics*, vol. 86, pp. 25–32, 2018.

[5] J. Kang and J. Gwak, "Ensemble of instance segmentation models for polyp segmentation in colonoscopy images," *IEEE Access*, vol. 7, pp. 26440–26447, 2019.

[6] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley Publishing, 2nd ed., 2014.

[7] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[8] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.

[9] T. Tajti, "New voting functions for neural network algorithms," *Annales Mathematicae et Informaticae*, vol. 52, pp. 229–242, 2020.

[10] M. Mohandes, M. Deriche, and S. O. Aliyu, "Classifiers combination techniques: A comprehensive review," *IEEE Access*, vol. 6, pp. 19626–19639, 2018.

[11] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, (Berlin, Heidelberg), pp. 1–15, Springer Berlin Heidelberg, 2000.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[13] S. Mirjalili, *Evolutionary Algorithms and Neural Networks*. Springer International Publishing, 2019.

[14] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method*. Springer New York, 2004.

[15] S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, 1993.

[16] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (New York, NY, USA), pp. 661–670, 2014.

[17] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010* (Y. Lechevallier and G. Saporta, eds.), (Heidelberg), pp. 177–186, Physica-Verlag HD, 2010.

[18] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, 2016.

[19] D. Masters and C. Luschi, "Revisiting small batch training for deep neural networks," *arXiv:1804.07612*, 2018.

[20] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: Closing the generalization gap in large batch training of neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), pp. 1729–1739, Curran Associates Inc., 2017.

[21] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't Decay the Learning Rate, Increase the Batch Size," *arXiv:1711.00489*, 2017.

[22] Z. Huo, B. Gu, and H. Huang, "Large batch optimization for deep learning using new complete layer-wise adaptive rate scaling," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 9, pp. 7883–7890, 2021.

[23] C. Darken, J. Chang, and J. Moody, "Learning rate schedules for faster stochastic gradient search," in *Neural Networks for Signal*

Processing II Proceedings of the 1992 IEEE Workshop, pp. 3–12, 1992.

[24] J. Tóth, H. Tomán, and A. Hajdu, "Improving the performance of an ensemble-based exudate detection system using stochastic parameter optimization," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5243–5246, 2015.

[25] J. Tóth, L. Szakács, and A. Hajdu, "Finding the optimal parameter setting for an ensemble-based lesion detector," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 3532–3536, Oct 2014.

[26] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[27] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *Handbook of Metaheuristics*, pp. 1–35, Springer International Publishing, 2018.

[28] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.

[29] H. J. Kushner, "Asymptotic global behavior for stochastic approximation and diffusions with slowly decreasing noise effects: Global minimization via Monte Carlo," *SIAM Journal on Applied Mathematics*, vol. 47, no. 1, pp. 169–185, 1987.

[30] S. B. Gelfand and S. K. Mitter, "Simulated annealing with noisy or imprecise energy measurements," *Journal of Optimization Theory and Applications*, vol. 62, no. 1, pp. 49–62, 1989.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 3rd ed., 2009.

[32] W. J. Gutjahr and G. C. Pflug, "Simulated annealing for noisy cost functions," *Journal of Global Optimization*, vol. 8, no. 1, pp. 1–13, 1996.

[33] R. Bethea, *Statistical Methods for Engineers and Scientists*. CRC Press, 3 ed., 2019.

[34] J. B. Jonas and C. Sabanayagam, "Epidemiology and risk factors for diabetic retinopathy," in *Frontiers in Diabetes*, pp. 20–37, S. Karger AG, 2019.

[35] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *Knowledge-Based Systems*, vol. 60, pp. 20–27, 2014.

[36] R. Biyani and B. Patre, "Algorithms for red lesion detection in diabetic retinopathy: A review," *Biomedicine & Pharmacotherapy*, vol. 107, pp. 681–688, 2018.

[37] S. Morales, K. Engan, V. Naranjo, and A. Colomer, "Retinal disease screening through local binary patterns," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 184–192, 2017.

[38] A. Colomer, J. Igual, and V. Naranjo, "Detection of early signs of diabetic retinopathy based on textural and morphological information in fundus images," *Sensors*, vol. 20, no. 4, p. 1005, 2020.

[39] K. Shankar, A. R. W. Sait, D. Gupta, S. Lakshmanaprabu, A. Khanna, and H. M. Pandey, "Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model," *Pattern Recognition Letters*, vol. 133, pp. 210–216, 2020.

[40] T. Araújo, G. Aresta, L. Mendonça, S. Penas, C. Maia, Â. Carneiro, A. M. Mendonça, and A. Campilho, "DR|GRADUATE: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images," *Medical Image Analysis*, vol. 63, p. 101715, 2020.

[41] J. Xu, X. Zhang, H. Chen, J. Li, J. Zhang, L. Shao, and G. Wang, "Automatic analysis of microaneurysms turnover to diagnose the progression of diabetic retinopathy," *IEEE Access*, vol. 6, pp. 9632–9642, 2018.

[42] J. H. Hipwell, F. Strachan, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester, "Automated detection of microaneurysms in digital red-free photographs: a diabetic retinopathy screening tool," *Diabetic Medicine*, vol. 17, no. 8, pp. 588–594, 2000.

[43] M. J. Cree, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester, "A fully automated comparative microaneurysm digital detection system," *Eye*, vol. 11, no. 5, pp. 622–628, 1997.

[44] A. D. Fleming, S. Philip, K. A. Goatman, J. A. Olson, and P. F. Sharp, "Automated microaneurysm detection using local contrast normalization and local vessel detection," *IEEE Transactions on Medical Imaging*, vol. 25, no. 9, pp. 1223–1232, 2006.

[45] A. Bhalerao, A. Patanaik, S. Anand, and P. Saravanan, "Robust detection of microaneurysms for sight threatening retinopathy screening," in *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pp. 520–527, 2008.

[46] L. Giancardo, T. P. Karnowski, K. W. Tobin, F. Meriaudeau, and E. Chaum, "Validation of microaneurysm-based diabetic retinopathy screening across retina fundus datasets," in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pp. 125–130, 2013.

[47] B. Harangi and A. Hajdu, "Exudate detection in fundus images using active contour methods and region-wise classification," in *Biomedical Image Segmentation: Advances and Trends*, pp. 157–186, CRC Press, 2019.

[48] S. Ravishankar, A. Jain, and A. Mittal, "Automated feature extraction for early detection of diabetic retinopathy in fundus images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 210–217, 2009.

[49] P. Buyssens, M. Daisy, D. Tschumperle, and O. Lezoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1809–1824, 2015.

[50] T. Walter and J.-C. Klein, "Automatic detection of microaneurysms in color fundus images of the human retina by means of the bounding box closing," in *Medical Data Analysis* (A. Colosimo, P. Sirabella, and A. Giuliani, eds.), (Berlin, Heidelberg), pp. 210–220, Springer Berlin Heidelberg, 2002.

[51] I. Lázár and A. Hajdu, "Microaneurysm detection in retinal images using a rotating cross-section based model," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1405–1409, 2011.

[52] T. Walter, P. Massin, A. Erginay, R. Ordonez, C. Jeulin, and J.-C. Klein, "Automatic detection of microaneurysms in color fundus images," *Medical Image Analysis*, vol. 11, no. 6, pp. 555–566, 2007.

[53] B. Zhang, X. Wu, J. You, Q. Li, and F. Karray, "Detection of microaneurysms using multi-scale correlation coefficients," *Pattern Recognition*, vol. 43, no. 6, pp. 2237–2248, 2010.

[54] B. Harangi, J. Tóth, and A. Hajdu, "Fusion of deep convolutional neural networks for microaneurysm detection in color fundus images," in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3705–3708, 2018.

[55] E. Decencière, G. Cazuguel, X. Zhang, G. Thibault, J.-C. Klein, F. Meyer, B. Marcotegui, G. Quellec, M. Lamard, R. Danno, D. Elie, P. Massin, Z. Viktor, A. Erginay, B. Laÿ, and A. Chabouis, "Teleophta: Machine learning and image processing methods for teleophthalmology," *IRBM*, vol. 34, no. 2, pp. 196 – 203, 2013. Special issue : ANR TECSAN : Technologies for Health and Autonomy.

[56] Kaggle Inc., "Diabetic Retinopathy Detection." `https://www.kaggle.com/c/diabetic-retinopathy-detection`, 2020. [Online; accessed on 19 January 2020].

[57] R. J. Chalakkal, W. H. Abdulla, and S. S. Thulaseedharan, "Quality and content analysis of fundus images using deep learning," *Computers in Biology and Medicine*, vol. 108, pp. 317–331, 2019.

[58] P. Eusebi, "Diagnostic accuracy measures," *Cerebrovascular Diseases*, vol. 36, no. 4, pp. 267–272, 2013.

[59] D. Chakraborty, *Observer Performance Methods for Diagnostic Imaging: Foundations, Modeling, and Applications with R-based Examples*. CRC Press, 2017.

[60] P. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, pp. 532–540, April 1983.

[61] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.

[62] H. R Buie, G. Campbell, R. Joshua Klinck, J. A MacNeil, and S. K Boyd, "Automatic segmentation of cortical and trabecular compartments based on a dual threshold technique for in vivo micro-CT bone analysis," *Bone*, vol. 41, pp. 505–515, 2007.

[63] H. Zhou, G. Schaefer, and C. Shi, *Fuzzy C-Means Techniques for Medical Image Segmentation*, ch. 13, pp. 257–271. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[64] D. Lim Fat, J. Kennedy, R. Galvin, F. O'Brien, F. Mc Grath, and H. Mullett, "The Hounsfield value for cortical bone geometry in the proximal humerus—an in vitro study," *Skeletal Radiology*, vol. 41, no. 5, pp. 557–568, 2012.

[65] M. Kardell, M. Magnusson, M. Sandborg, G. Alm Carlsson, J. Jeuthe, and A. Malusek, "Automatic segmentation of pelvis for brachytherapy of prostate," *Radiation Protection Dosimetry*, vol. 169, no. 1-4, pp. 398–404, 2016.

[66] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[67] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 100–132, 1985.

[68] W. R. Crum, O. Camara, and D. L. G. Hill, "Generalized overlap measures for evaluation and validation in medical image analysis," *IEEE Transactions on Medical Imaging*, vol. 25, pp. 1451–1461, 2006.

[69] J. K. Leader, B. Zheng, R. M. Rogers, F. C. Sciurba, A. Perez, B. E. Chapman, S. Patel, C. R. Fuhrman, and D. Gur, "Automated lung segmentation in X-ray computed tomography," *Academic Radiology*, vol. 10, no. 11, pp. 1224–1236, 2003.

[70] R. Bellotti, F. De Carlo, G. Gargano, S. Tangaro, D. Cascio, E. Catanzariti, P. Cerello, S. C. Cheran, P. Delogu, I. De Mitri, C. Fulcheri, D. Grosso, A. Retico, S. Squarcia, E. Tommasi, and B. Golosio, "A CAD system for nodule detection in low-dose lung CTs based on region growing and a new active contour model," *Medical Physics*, vol. 34, no. 12, pp. 4901–4910, 2007.

[71] E. E. Nithila and S. Kumar, "Segmentation of lung from CT using various active contour models," *Biomedical Signal Processing and Control*, vol. 47, pp. 57–62, 2019.

[72] A. Halder, S. Chatterjee, D. Dey, S. Kole, and S. Munshi, "An adaptive morphology based segmentation technique for lung nodule detection in thoracic CT image," *Computer Methods and Programs in Biomedicine*, vol. 197, p. 105720, 2020.

[73] S. Raj, D. S. Vinod, B. S. Mahanand, and N. Murthy, "Intuitionistic fuzzy c means clustering for lung segmentation in diffuse lung diseases," *Sensing and Imaging*, vol. 21, no. 1, 2020.

[74] J. Hofmanninger, F. Prayer, J. Pan, S. Röhrich, H. Prosch, and G. Langs, "Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem," *European Radiology Experimental*, vol. 4, no. 1, 2020.

[75] J. Tan, L. Jing, Y. Huo, L. Li, O. Akin, and Y. Tian, "LGAN: Lung segmentation in CT scans using generative adversarial network," *Computerized Medical Imaging and Graphics*, vol. 87, p. 101817, 2021.

[76] M. Jun, G. Cheng, W. Yixin, A. Xingle, G. Jiantao, Y. Ziqi, Z. Minqing, L. Xin, D. Xueyuan, C. Shucheng, and et al., "COVID-19 CT Lung and Infection Segmentation Dataset," April 2020.

[77] RAIOSS, "Coronacases.org by RAIOSS.com." `https://coronacases.org/`. [Online; accessed on 20 March 2022].

[78] Radiopedia, "COVID-19: Radiology reference article." `https://radiopaedia.org/articles/covid-19-4?lang=us`. [Online; accessed on 20 March 2022].

Summary

In this dissertation, three methods are presented for the efficient parameter optimization of ensembles in medical image analysis. The development of these methods was motivated by the fact that the individually optimal parameter settings of the members do not necessarily maximize the performance of the ensemble. Therefore, system-level parameter optimization is required, which can lead to large-scale problems. Stochastic methods can be used to find good quality solutions to such problems, sacrificing some accuracy to significantly reduce the cost of the search. However, even a stochastic search can be very expensive if the evaluation of a solution itself is expensive.

The methods proposed in this dissertation are based on the metaheuristic simulated annealing and aim to reduce the cost of the search by determining the value of the objective function, i.e., the performance of the ensemble with a given parameter setting, using partial training data. This approach can be considered as a form of noisy evaluation with imprecise measurements. By employing different strategies, the proposed methods are able to control the noise during the search process according to theoretical constraints and thus maintain the achievable solution quality.

The first method presented was developed to accelerate the parameter optimization of ensembles on large datasets. It employs a

sampling-based evaluation method that considers only a randomly selected subset of the training data with the minimum required cardinality in each iteration to reduce the cost of the search. The sample sizes required during the search process are theoretically determined by adapting convergence results for noisy evaluation in simulated annealing. The effectiveness of the method was demonstrated using the parameter optimization problem of two ensembles that classify retinal images according to the presence of diabetic retinopathy.

As an alternative approach to accelerate the parameter optimization of ensembles using partial training data, the second method presented in the dissertation uses increasingly higher resolution levels of a pyramid representation of the images in the training set to evaluate solutions as the search progresses. To ensure that the convergence conditions are met, a strategy was proposed to determine the highest level (lowest image resolution) that can be used in each iteration to control the noise. The applicability and efficiency of this method were shown through the parameter optimization problem of an ensemble for bone segmentation on computed tomography images.

The third method presented is based on the observation that, depending on the optimization problem, image downscaling can introduce noise with a lower standard deviation than dataset sampling with the same cost gain. Therefore, it employs an evaluation method that combines sampling of the training data with image downscaling to further accelerate the parameter optimization of ensembles. To this end, a strategy was proposed that determines the appropriate scaling level and sample size in each iteration by adapting the convergence results for noisy evaluation. Using the parameter optimization problem of an ensemble that segments the lungs in computed tomography scans, it was shown that this method allows further reduction of the cost of the search while maintaining solution quality compared to the previous

two methods when the size of the dataset is below a problem-specific value.

Összefoglaló

Ebben az értekezésben három módszer kerül bemutatásra algoritmus-együttesek paraméteroptimalizálásának hatékony elvégzésére az orvosi képelemzés területén. E módszerek kidolgozását az motiválta, hogy az együtteseket alkotó tagok egyedileg optimális paraméterbeállításai nem szükségszerűen maximalizálják a rendszer teljesítményét. Emiatt rendszerszintű paraméteroptimalizálásra van szükség, ami nagyméretű problémákat eredményezhet. Ilyen problémák esetében jó minőségű megoldásokat lehet találni sztochasztikus módszerekkel, némi pontosságot feláldozva a keresés költségének jelentős csökkentése érdekében. Azonban még egy sztochasztikus módszerrel végzett keresés is lehet nagyon költséges, ha az egyes megoldások kiértékelése önmagában is költséges.

Az értekezésben javasolt módszerek a szimulált hűtés metaheurisztikán alapulnak, és céljuk a keresés költségének csökkentése azáltal, hogy a célfüggvény értékét, azaz a rendszer teljesítményét egy adott paraméterbeállítás mellett, részleges tanulóadatok használatával határozzák meg. Ez a megközelítés egyfajta zajos, pontatlan mérésekkel végzett kiértékelésnek tekinthető. A javasolt módszerek különböző stratégiák alkalmazásával képesek a keresési folyamat során az elméleti megkötéseknek megfelelően szabályozni a zajt, és így fenntartani az elérhető megoldás minőségét.

Az első bemutatott módszer algoritmusegyüttesek nagyméretű adathalmazok felett történő paraméteroptimalizálásának gyorsítására került kidolgozásra. Ez egy olyan mintavételezésen alapuló kiértékelési módszert alkalmaz, amely minden iterációban a tanulóadatoknak csak egy minimálisan szükséges elemszámú, véletlenszerűen kiválasztott részhalmazát tekinti annak érdekében, hogy csökkentse a keresés költségét. A keresési folyamat során szükséges mintaméretek elméleti úton kerülnek meghatározásra a zajos kiértékelést használó szimulált hűtésre vonatkozó konvergenciaeredmények adaptálásával. A módszer hatékonysága két olyan algoritmusegyüttes paraméteroptimalizálási problémáján keresztül lett megmutatva, amelyek szemfenékfelvételeket osztályoznak a diabéteszes retinopátia jelenléte alapján.

Egy másik megközelítésként az algoritmusegyüttesek paraméteroptimalizálásának részleges tanulóadatok használatával történő gyorsítására, az értekezésben bemutatott második módszer a keresés előrehaladtával a megoldások kiértékeléséhez a tanulóhalmazbeli képek egy piramis reprezentációjának egyre magasabb felbontású szintjeit használja. A konvergenciafeltételek teljesülésének biztosításához egy olyan stratégia lett javasolva, amely alkalmas az egyes iterációkban használható legmagasabb szint (legalacsonyabb felbontás) meghatározására, és ezáltal a zaj szabályozására. E módszer alkalmazhatósága és hatékonysága egy komputertomográfiai felvételeken történő csontszegmentálásra kidolgozott algoritmusegyüttes paraméteroptimalizálási problémáján keresztül lett megmutatva.

A harmadik bemutatott módszer azon a megfigyelésen alapul, hogy az optimalizálási problémától függően, a képek leskálázása kisebb szórású zajt is eredményezhet, mint az adathalmaz mintavételezése azonos költségbeli nyereség mellett. Ezért egy olyan kiértékelési módszert alkalmaz, amely a tanulóadatok mintavételezését kombinálja a képek leskálázásával, annak érdekében, hogy tovább csökkentse a paraméterop-

timalizálás időigényét. Ehhez egy olyan stratégia lett javasolva, amely az egyes iterációkban szükséges leskálázási szintet és mintaméretet a zajos kiértékelésre vonatkozó konvergenciaeredmények adaptálásával határozza meg. Egy, a tüdő komputertomográfiai felvételeken történő szegmentálására szolgáló algoritmusegyüttes paraméteroptimalizálási problémáján keresztül meg lett mutatva, hogy a módszer lehetővé teszi a keresés költségének további csökkentését a megoldás minőségének megőrzése mellett az előző két módszerhez hasonlítva, ha az adathalmaz elemszáma kisebb, mint egy problémaspecifikus érték.

List of Publications

Journal Articles in English

[P1] J. Tóth, H. Tomán, A. Hajdu, *Using Noisy Evaluation to Accelerate Parameter Optimization with Simulated Annealing*, Computers and Electrical Engineering, 2022. (SJR: Q1, IF: 4.152) (*submitted*)

[P2] X. Huang, S. Zhou, J. Tóth, A. Hajdu, *Cuproptosis-related gene index: a predictor for pancreatic cancer prognosis, immunotherapy efficacy, and chemosensitivity*, Frontiers in Immunology, vol. 13, 978865, 2022. (SJR: Q1, IF: 8.786) (*accepted for publication*) doi:10.3389/fimmu.2022.978865

[P3] G. Bogacsovics, J. Tóth, A. Hajdu, B. Harangi, *Enhancing CNNs Through the Use of Hand-crafted Features in Automated Fundus Image Classification*, Biomedical Signal Processing and Control, vol. 76, 103685, 2022. (SJR: Q1, IF: 5.076) doi:10.1016/j.bspc.2022.103685

[P4] J. Tóth, H. Tomán, A. Hajdu, *Efficient Sampling-based Energy Function Evaluation for Ensemble Optimization Using Simulated Annealing*, Pattern Recognition, vol. 107, 107510, 2020. (SJR: D1, IF: 7.740) doi:10.1016/j.patcog.2020.107510

[P5] P. Porwal *et al.* (J. Tóth, Á. Baran, B. Harangi, A. Hajdu), *IDRiD: Diabetic Retinopathy – Segmentation and Grading Challenge*, Medical Image Analysis, vol. 59, 101561, 2020. (SJR: D1, IF: 8.545) doi:10.1016/j.media.2019.101561

[P6] J. Tóth, R. Tornai, I. Labancz, A. Hajdu, *Efficient Visualization for an Ensemble-based System*, Acta Polytechnica Hungarica, vol. 16, no. 2, pp. 59-75, 2019. (SJR: Q2, IF: 1.219) doi:10.12700/APH.16.2.2019.2.4

[P7] B. Antal, M. K. G. S. Tavares, L. Kovács, B. Harangi, I. Lázár, B. Nagy, Gy. Kovács, J. Szakács, J. Tóth, T. Pető, A. Csutak, A. Hajdu, *Data Analysis Applied to Diabetic Retinopathy Screening: Performance Evaluation*, Annales Mathematicae et Informaticae, vol. 49, pp. 3-9, 2018. (SJR: Q3) doi:10.33039/ami.2018.10.002

[P8] R. Besenczi, J. Tóth, A. Hajdu, *A Review on Automatic Analysis Techniques for Color Fundus Photographs*, Computational and Structural Biotechnology Journal, vol. 14, pp. 371-384, 2016. (SJR: Q1) doi:10.1016/j.csbj.2016.10.001

Journal Articles in Hungarian

[P9] G. Bogacsovics, A. Hajdu, B. Harangi, I. Lakatos, R. Lakatos, M. Szabó, A. Tiba, J. Tóth, *Napelemfarmok Magyarország területén történő elhelyezését segítő döntéstámogató rendszer fejlesztése*, Közigazgatás-tudomány, vol. 1, no. 2, pp. 134-145, 2021. doi:10.54200/kt.v1i2.23

[P10] G. Bogacsovics, A. Hajdu, B. Harangi, I. Lakatos, R. Lakatos, M. Szabó, A. Tiba, J. Tóth, Á. Tarcsi, *Adatelemzési folyamat és keretrendszer a közigazgatás számára*, Közigazgatás-tudomány, vol. 1, no. 2, pp. 146-158, 2021. doi:10.54200/kt.v1i2.24

Conference Papers

[P11] J. Tóth, T. P. Kapusi, B. Harangi, H. Tomán, A. Hajdu, *Accelerating the Optimization of a Segmentation Ensemble using Image Pyramids*, 11$^{\text{th}}$ International Symposium on Image and Signal Processing and Analysis (ISPA 2019), Dubrovnik, Croatia, 23-25 September 2019, pp. 43-48. doi:10.1109/ISPA.2019.8868860

[P12] B. Harangi, J. Tóth, G. Bogacsovics, D. Kupás, L. Kovács, A. Hajdu, *Cell Detection on Digitized Pap Smear Images using Ensemble of Conventional Image Processing and Deep Learning Techniques*, 11$^{\text{th}}$ International Symposium on Image and Signal Processing and Analysis (ISPA 2019), Dubrovnik, Croatia, 23-25 September 2019, pp. 38-42. doi:10.1109/ISPA.2019.8868683

[P13] B. Harangi, J. Tóth, Á. Baran, A. Hajdu, *Automatic Screening of Fundus Images Using a Combination of Convolutional Neural Network and Hand-Crafted Features*, 41$^{\text{st}}$ Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2019), Berlin, Germany, 23-27 July 2019, pp. 2699-2702. doi:10.1109/EMBC.2019.8857073

[P14] B. Harangi, J. Tóth, A. Hajdu, *Fusion of Deep Convolutional Neural Networks for Microaneurysm Detection in Color Fundus Images*, 40$^{\text{th}}$ Annual International Conference of the

IEEE Engineering in Medicine and Biology Society (EMBC 2018), Honolulu, HI, USA, 17-21 July 2018, pp. 3705-3708. doi:10.1109/EMBC.2018.8513035

[P15] J. Tóth, L. Bartha, T. Szabó, I. Lázár, B. Harangi, A. Hajdu, *An Online Application for Storing, Analyzing, and Sharing Dermatological Data*, IEEE 6th International Conference on Cognitive Infocommunications (CogInfoCom 2015), Győr, Hungary, 19-21 October 2015, pp. 339-342. doi:10.1109/CogInfoCom.2015.7390615

[P16] J. Tóth, H. Tomán, A. Hajdu, *Improving the Performance of an Ensemble-based Exudate Detection System Using Stochastic Parameter Optimization*, 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015), Milan, Italy, 25-29 August 2015, pp. 5243-5246. doi:10.1109/EMBC.2015.7319574

[P17] J. Tóth, L. Kovács, B. Harangi, Cs. Kiss, A. Mohácsi, Z. Orosz, A. Hajdu, *An Online Benchmark System for Image Processing Algorithms*, IEEE 5th International Conference on Cognitive Infocommunications (CogInfoCom 2014), Vietri sul Mare, Italy, 5-7 November 2014, pp. 377-382. doi:10.1109/CogInfoCom.2014.7020482

[P18] J. Tóth, L. Szakács, A. Hajdu, *Finding the Optimal Parameter Setting for an Ensemble-based Lesion Detector*, IEEE 21st International Conference on Image Processing (ICIP 2014), Paris, France, 27-30 October 2014, pp. 3532-3536. doi:10.1109/ICIP.2014.7025717

[P19] J. Tóth, I. Papp, R. Tornai, I. Labancz, E. Hajduné Pocsai, A. Hajdu, *Cognitive Visualization for the Design of Complex Systems*, IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom 2013), Budapest, Hungary, 2-5 December 2013, pp. 363-368. doi:10.1109/CogInfoCom.2013.6719272

[P20] Cs. Lámfalusi, D. Girus, K. Kruppa, J. Tóth, E. Hajduné Pocsai, R. Kunkli, A. Hajdu, L. B. Bálint, *Extending the Visualization Capabilities of a Genome Browser*, IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom 2013), Budapest, Hungary, 2-5 December 2013, pp. 419-422. doi:10.1109/CogInfoCom.2013.6719283

[P21] A. Hajdu, J. Tóth, Z. Pistár, B. Domokos, Zs. Török, *An Ensemble-based Collaborative Framework to Support Customized User Needs*, IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom 2012), Kosice, Slovakia, 2-5 December 2012, pp. 285-290. doi:10.1109/CogInfoCom.2012.6421995

Conference Abstracts

[P22] A. Hajdu, B. Harangi, J. Tóth, M. Pap, *Combining Convolutional Neural Networks and Hand-Crafted Features in Medical Image Classification Tasks*, 20th European Conference on Mathematics for Industry, Budapest, Hungary, 18-22 June 2018, p. 299.

[P23] J. Tóth, L. Kovács, B. Harangi, Cs. Kiss, A. Mohácsi, Z. Orosz, A. Hajdu, *An Online System for Algorithm Benchmarking*, IEEE

5th International Conference on Cognitive Infocommunications (CogInfoCom 2014), Vietri sul Mare, Italy, 5-7 November 2014, pp. 383. doi:10.1109/CogInfoCom.2014.7020483

[P24] Cs. Lámfalusi, D. Girus, K. Kruppa, J. Tóth, E. Hajduné Pocsai, R. Kunkli, A. Hajdu, L. B. Bálint, *Adding a Scalable Visualization Technique to the UCSC Genome Browser*, IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom 2013), Budapest, Hungary, 2-5 December 2013, pp. 943.