

Autonomous University of State of Mexico
University Centre UAEM Texcoco



Development of evolutionary algorithms for
automatic image segmentation

As a fulfillment requirement to obtain
Master degree in Computer Science

Submitted by
Jared Cervantes Canales

Thesis advisor
Dr. Farid García Lamont

Texcoco, State of Mexico, October 2021

Abstract

A vision system is made up of several steps, each of which influences the final result. One of those steps is segmentation; this allows us to obtain only the region of interest from the object and eliminate the background. Segmentation is essential because it allows to improve the quality of the segmented region, improve the quality of the extracted features and at the same time reduce the noise introduced by the poor quality of features. There are various segmentation techniques in the literature, each with one or more variables to tune. Choosing the appropriate technique to segment a specific data set is not an easy task since, on some occasions, various factors can influence the quality of the segmentation, such as pre-processing and data set. In addition to the above, slight variations in lighting and intensity changes can influence the segmentation quality. Sometimes an expert is necessary to select the segmentation technique and fine-tune the associated variables. In this Thesis, automatic algorithms are developed for the selection of segmentation techniques and associated variables. The techniques developed are implemented and compared using various data sets. The experimental results obtained are discussed and compared in this Thesis. These algorithms will facilitate and simplify the selection of the segmentation techniques to be used and the necessary variables and implemented pre-processing.

Resumen

Un sistema de visión esta compuesto por varios pasos, cada uno de ellos influye en el resultado final. Uno de esos pasos es la segmentación, esta permite obtener solo la región que nos interesa del objeto y eliminar el fondo. La segmentación es muy importante pues permite mejorar la calidad de la región segmentada, mejorar la calidad de las características extraídas y al mismo tiempo reducir el ruido introducido por una mala calidad de características. Existen diversas técnicas de segmentación en la literatura, cada una de ellas con una o varias variables para afinar. Elegir la técnica adecuada para segmentar un determinado conjunto de datos no es una tarea fácil, ya que en algunas ocasiones diversos factores pueden influir en la calidad de la segmentación como pre-procesamiento y conjunto de datos. Aunado a lo anterior, pequeñas variaciones en la iluminación y cambios de intensidad pueden influir demasiado en la calidad de la segmentación. En algunas ocasiones es necesario un experto para seleccionar la técnica de segmentación y poner a punto las variables asociadas. En esta Tesis, se desarrollan algoritmos automáticos para selección de técnicas de segmentación y variables asociadas. Las técnicas desarrolladas son implementadas y comparadas utilizando varios conjuntos de datos. Los resultados experimentales obtenidos son discutidos y comparados en esta Tesis. Estos algoritmos permitirán facilitar y simplificar la selección de la técnicas de segmentación a utilizar, así como las variables necesarias y pre-procesamiento implementado.

Contents

Contents	xi
List of Figures	xv
1 Introduction	1
1.1 Problem statement	3
1.2 Justification	4
1.3 General objective	4
1.4 State of the art	4
1.4.1 Threshold-based techniques	5
1.4.2 Clustering-based techniques	6
1.4.3 Techniques based on edge detection	7
1.4.4 Region-based techniques	7
1.4.5 Techniques based on neural networks	8
1.5 Thesis organization	10
2 Preliminaries	11
2.1 Segmentation techniques	11
2.1.1 Threshold-based segmentation techniques	14
Global Threshold	14
Adaptive threshold	14
Otsu method	16
2.1.2 Techniques based on edge detection	18
Border detection	18
Gradient method	19
Laplacian	20
2.1.3 Clustering-based methods	20
K-means	21
2.1.4 Region-Based Methods	21

	Split and Merge	22
2.1.5	AI-Based Methods	22
	Neural networks	22
	Support vector machines	23
2.2	Color spaces	23
2.2.1	RGB space color	23
2.2.2	HSV space color	24
2.2.3	HSI space color	25
2.3	Genetic algorithms	26
	Elements of a genetic algorithm.	26
2.3.1	Basic genetic Algorithm	27
2.3.2	Initial population	27
2.3.3	Selection	28
2.3.4	Crossover	29
2.3.5	Mutation	29
2.3.6	Stop condition	30
3	Metodology	31
3.1	Datasets	34
3.2	Image selection	35
3.2.1	Sefexa Image Segmentation Tool	35
3.3	Selection of techniques and parameters	37
3.3.1	Initial population and chromosome representation	38
	Case 1 (9 variables)	39
	Case 2 (11 variables)	40
3.3.2	Fitness function assessment	42
	Probabilistic Rand Index (PRI)	42
	Global Consistency Error (GCE)	43
	Variation of Information (VOI)	43
3.3.3	Selection	44
3.3.4	Crossover	45
3.3.5	Mutation	45
3.3.6	Stop Condition	46
3.3.7	Elitism	46

4	Experimental results	47
4.1	Data sets	47
4.1.1	ICL Data set	47
4.1.2	Coffee diseases data set	48
4.1.3	PlantVillage data set	49
4.1.4	LSM data set	50
4.2	Segmentation of data sets	51
4.2.1	ICL data set	51
4.2.2	Coffee diseases data set	52
4.2.3	PlantVillage data set	53
4.2.4	LSM data set	54
4.3	Results and comparative analysis	55
4.3.1	Convergence of the genetic algorithm	55
4.4	Discusions	56
4.4.1	Results obtained with the Coffee Diseases data set	57
4.4.2	Results obtained with the PlantVillage dataset	60
4.4.3	Results obtained with the LSM data set	62
5	Conclusions	67
	Bibliography	69

List of Figures

1.1	Papers in state of the art	9
2.1	Semantic segmentation [Taken from https://informatica.blogs.uoc.edu/la-segmentacion-semantic-y-sus-benchmarks/]	12
2.2	Segmentation of instances [Taken from https://informatica.blogs.uoc.edu/la-segmentacion-semantic-y-sus-benchmarks/]	12
2.3	Elemental segmentation methods	13
2.4	Laplacian convolutional filter	20
2.5	RGB space color model	24
2.6	HSV spcae color model [Taken from https://www.researchgate.net/figure/Figura-23-Espacio-de-colores-HSV_fig2_326551993]	25
2.7	Graphic representation of expressions used in GA	27
2.8	Crossover operator	29
2.9	Mutation operator	29
3.1	Component of a vision system	31
3.2	Techniques and segmentation parameters	32
3.3	Proposed metodology	33
3.4	Datasets used in the experimental results	34
3.5	Selection of the set of images and obtaining of segmentation parameters	36
3.6	Sefexa software interface	36
3.7	Segmentation by sefexa software	37
3.8	Genetic algorithm flow chart	38
3.9	Chromosomal chain of nine variables	39
3.10	Chromosomal chain of eleven variables	40
3.11	Representation of fitness percentage	44
3.12	Two points crossover	45
4.1	ICL Dataset	48
4.2	Coffee diseases data set	49

4.3	PlantVillage dataset	50
4.4	LSM dataset	51
4.5	Segmentation of ICL data set	52
4.6	Segmentation of Coffee diseases data set	53
4.7	Segmentation of PlantVillage data set	54
4.8	Segmentation of LSM dataset	55
4.9	Average distance between individuals	56
4.10	Best individual and average fitness of individuals	56
4.11	The Probabilistic Rand Index (PRI) for each image in Coffee diseases dataset	57
4.12	The Global Consistency Error (GCE) for each image in Coffee diseases dataset	58
4.13	The Variance of Information (VoI) for each image in Coffee diseases dataset	59
4.14	The Probabilistic Rand Index (PRI) for each image in PlantVillage dataset	60
4.15	The Global Consistency Error (GCE) for each image in PlantVillage dataset	61
4.16	The Variance of Information (VoI) for each image in PlantVillage dataset	62
4.17	The Probabilistic Rand Index (PRI) for each image in LSM dataset	63
4.18	The Global Consistency Error (GCE) for each image in LSM dataset	64
4.19	The Variance of Information (VoI) for each image in LSM dataset	65

Chapter 1

Introduction

The development of automatic vision and image processing systems has been of great interest to solve real-life problems. Many areas such as medicine [GSJ20] [DTV18] [Xia+18], civil engineering [Tao+20], Astronomy [Rui+18], security [TKR20], robotics [KS19], and agriculture [CM+20] support each other and make use of image processing techniques, genes, and vision to process and analyze information that is used to make decisions.

At present, even though digital image processing systems and vision systems have made great progress, they present several problems when implemented. Vision systems depend on several steps, and each step is carried out by some technique that depends on its variables. In general, a vision system consists of pre-processing, segmentation, extraction of characteristics, selection of characteristics, and classification, among others. One of the most important steps in a vision system is segmentation. Segmentation is a very important and very sensitive step in a vision system. Small changes in the pre-processing, variables of each technique, lighting or intensity of the images, or even morphological operations of dilation or erosion to polish the final image can cause poor segmentation. The main objective of this work is the development of automatic algorithms for the segmentation of image sets.

Segmentation is defined as the process of dividing the image into multiple regions. The most basic segmentation is binary segmentation. It divides the image into only two regions, the region of interest and the background of the image. This process facilitates the image analysis by sectioning and obtaining its most representative part and eliminating the one that does not represent interest.

Formally, the segmentation of images of genes is defined as [GL+18]:

$$I_{segmented} = \cup_{i=1}^n Ri \quad (1.1)$$

Such that:

$$\bigcap_{i=1}^n Ri \neq \emptyset \quad (1.2)$$

Where n represents the number of segments

$$R_k = \{(i, j) \in N^2 \mid I(i, j) = \delta_k\} \quad (1.3)$$

$I(i, j)$ is the value of the local pixel in (i, j) of the input image I and δ_k is the threshold value in the k place.

In other words, it denotes a process by which the input image is divided into non-overlapping regions. Each region is homogeneous and connected. Each region in a segmented image must satisfy the properties of homogeneity and connectivity. The region is considered homogeneous if all the pixels in the region satisfy the homogeneity conditions defined by one or more pixel attributes, such as intensity, color, texture, etc.

There are various segmentation techniques in the literature, each with one or more variables to refine. Choosing the proper method to segment a specific data set is not an easy task. Multiple factors can influence the quality of segmentation, such as pre-processing techniques, variations in lighting, intensity changes, etc. Sometimes an expert is required to evaluate all aspects of the image set and perform experiments to select the appropriate segmentation technique and fine-tune the associated variables. However, the calculation of such variables for correct image segmentation has a very high combinatorial complexity. In addition, there is no general solution for all cases of segmentation. That is why, due to the huge solution space, evolutionary algorithms are necessary and justified.

In recent years evolutionary algorithms have become an essential tool for solving optimization problems. Optimization is seen as a decision problem that involves finding the best solution within a finite set of possible solutions [Huy+20] [JV20].

Optimization techniques and concepts are not limited to any discipline in particular and play an increasingly important role in solution and modeling in various fields of science ranging from bioinformatics, economics, structural design to modeling and automation.

Optimization systems are an important challenge in digital image processing. With particular relevance has been used in image processing problems, including edge detection, facial recognition, image segmentation, image compression, and feature extraction in Images.

In this Thesis, we developed automatic algorithms for the selection of segmentation techniques and associated variables. The techniques developed are implemented and compared using various data sets. In this introductory chapter, we explain the justification, describe the problem statement, define the general and specific objectives, a summary of the previous contributions, and finally describe the organization of the rest of the document.

1.1 Problem statement

A world where specific devices can detect objects, anomalies in a set of images, people or animals, gestures or signs in images is a reality lived in the present. Detecting objects has been a multi-process task that has been implemented in various fields of application. The basic process involved in this work is focused on the segmentation of images for subsequent extraction of characteristics.

However, identifying the right processes for optimal segmentation is not an easy process. It requires, in some cases, a deep understanding of each process involved. A correct obtaining of the variables involved and the techniques necessary for a correct segmentation is only obtained through years of experience and, in addition to this, multiple previous experiments to arrive at an adequate solution. That is why the development of automatic segmentation techniques is essential. In addition to the above, obtaining the processes necessary for optimal segmentation is not easy since the combination of processes is enormous. Also, there is no one-size-fits-all solution for targeting. That is why, due to the huge solution space, evolutionary algorithms are necessary and justified.

A genetic algorithm is developed to find the correct segmentation in a specific set of images from many individuals in this work. Each individual in our population is a possible solution to the problem of poor segmentation. Also, the individual is made up of different variables where they may contain preprocessing techniques, morphological techniques, color spaces, channels, and segmentation techniques. The job of the genetic algorithm is to evaluate each individual, select the best and make a cross and mutation of each one of them to improve our generation.

Testing each of these process combinations, and finding the most suitable one, is vital to obtain the appropriate feature extraction and therefore achieve automatic systems with better performance and capable of being implemented in real-life situations.

1.2 Justification

The development of automatic segmentation algorithms could improve the segmentation of image sets under similar lighting and background conditions. These algorithms will simplify the selection of the segmentation techniques to be used, the necessary variables and implemented pre-processing. To obtain an adequate model for the segmentation of sets of images, first is necessary to understand the influence of each step in the segmentation. To understand how the variables influence each one in particular and in combination with the get a segmentation.

Image segmentation is an essential process in the digital processing of images and vision systems whose study in recent years has been accentuated to solve multiple problems in the literature. The correct segmentation of images directly influences the performance of a vision system. That is why the study, development of new algorithms and their application is of utmost importance.

1.3 General objective

The general objective of this thesis is to develop algorithms for the automatic segmentation of image sets based on evolutionary algorithms.

It is intended to achieve the following specific objectives:

1. Analyze and identify processes that influence image segmentation.
2. Determine segmentation techniques to use, as well as parameters to use for each segmentation technique
3. Implement automatic segmentation algorithms based on genetic algorithms.

1.4 State of the art

In recent years there has been a great advance in developing image segmentation algorithms using evolutionary algorithms (EA). The numerous techniques developed can be classified according to the following categories:

1. Segmentation with EA using threshold-based techniques
2. Segmentation with EA using techniques based on Clustering
3. Segmentation with EA using techniques based on edge detection
4. Segmentation with EA using region-based techniques
5. Segmentation with EA using techniques based on neural networks

1.4.1 Threshold-based techniques

Threshold-based techniques are widely used in many applications because of their ease of implementation. In these algorithms, the main problem is to find the optimal threshold using histogram analysis. Since this task is usually computationally very expensive, different techniques have been used from very basic to more complex techniques aided by evolutionary algorithms [BY11] [OY18] [SB21] [Man+14]. The proposed algorithms can obtain a border (only desired region and background) or a multilevel border (different desired objects and background). The computation time and complexity increase exponentially with increasing the number of thresholds.

Banimelhem and Yahya [BY11] use a genetic algorithm to find optimal segmentation thresholds. In this article, the authors use a fitness function that evaluates the variance between objects in the image and the search is performed on a chromosome of size $L \times n$, where L is the number of intensities and n the number of thresholds used.

Oliveira and Yamanaka [OY18] They use a genetic algorithm to obtain the optimal frontier using Otsu's algorithm [Ots79]. The authors use intraclass variance as a fitness function and a chromosome size defined by the number of intensities. The authors compare their results with the Otsu method with one threshold and two thresholds.

Similarly, [Man+14] employs a genetic algorithm for the selection of the optimal threshold, peaks and valleys of the histogram. The authors argue that this new approach improves segmentation at a lower computational cost than other methods such as Otsu, having a normalized histogram function as the stop criterion. It is shared by [BM03], the difference is that in the first one, they add a transform to reduce the size of histograms and thus accelerate the convergence of the genetic algorithm.

In [PB15] the authors propose a segmentation algorithm for binary images containing circular and elliptical objects that have been artificially affected with different types of noise to test the segmentation quality. The proposed algorithm allows selecting morphological operators.

The authors in [PZL09] use a differential evolution algorithm to find the maximum variance between classes, that is, the optimal threshold of the Otsu algorithm [Ots79]. The authors perform a pre-processing step to reduce noise in the image.

1.4.2 Clustering-based techniques

Clustering techniques are essential methods in Machine learning. These techniques are called unsupervised learning techniques and allow obtaining subgroups of objects or data based on a degree of similarity or dissimilarity. Objects that are very similar to each other are grouped or labelled within a group, while very different objects are grouped or labelled in different groups. The algorithms used to clustering maximize the similarity between data belonging to the same class and minimize the similarity between data belonging to different classes.

The implementation of genetic algorithms, as support in clustering, has drastically improved the advances in a correct segmentation [Abb+11] [SMP11] [DBD16] [AS13].

In [SMP11] the authors present a fuzzy clustering technique based on multiobjective differential evolution. The authors argue that techniques that optimize a single target do not achieve good segmentation results. In their results, they compare the proposed technique with techniques that optimize a single objective. The proposed technique encodes the cluster centres and optimizes multiple validity measures simultaneously.

In [Abb+11] the authors use three k-means segmentation methods; the job of the genetic algorithm is to locate the colour regions in the HSI space to identify all the desired and unwanted pixels. This solution is assigned the name of GAHSI, which is the option to use a genetic algorithm in a single color space, the HSI (initials of hue -tone-, saturation -saturation- and intensity -intensity-).

Some proposals like [DBD16] describe a method where the chromosomes are formulated in a hierarchical structure to find the exact number of segments in a test image automatically. The genetic algorithm is used in conjunction with the Fuzzy c-means (FCM) algorithm to find the correct number of segments and automatic segmentation of abnormal and normal brain MRI images. In [AS13] the authors propose a new method that combines the genetic algorithm and the K-Means algorithm to segment medical images. In this combined technique, a variable chain length genetic algorithm is used to determine the optimal centres of the segmented regions.

More recently, [MEKS20] and [CM20] have proposed segmentation algorithms based on evolutionary techniques. In [MEKS20] the authors propose a local search algorithm for automatic clustering, which does not require a priori knowledge of the number of

clusters. The algorithm is able to find the number of clusters and their configuration automatically.

Chakraborty and Mali [CM20] propose a segmentation method based on Fuzzy c-means to improve the segmentation efficiency of biomedical images. The algorithm uses fuzzy membership functions to locate the optimal positions of the cluster centres. The approach does not depend on the starting positions of the chosen cluster centres.

1.4.3 Techniques based on edge detection

Local changes in intensity obtain edges in an image. Edge detection is the process of finding intensity changes in an image. Edge detection can be used as a segmentation technique when used with other methods.

Edge-based segmentation techniques work well on images with small intensity variation; however, the segmentation may not be good on images that have high variation. Some methods have been proposed in order to deal with this problem. In [DD20] the authors propose an algorithm based on Particle Swarm Optimization (PSO) that allows improving edge detection compared to classical techniques such as Canny and Prewitt.

Kumar and Raheja [KR20] propose an optimization method based on Ant Colony Optimization (ACO). The method transforms the input image's intensity values into pheromone values that are used to detect the edges in the image.

Diaz et al [DBB] propose a segmentation algorithm based on a genetic algorithm. The genetic algorithm is used to reduce false edges from thin and closed edges. According to the authors, the algorithm is sensitive to noise.

1.4.4 Region-based techniques

These techniques work in a similar way to clustering techniques. However, they are not the same. Unlike clustering techniques, region-based techniques generally use two steps. In the first step, the image is recursively divided using a homogeneity criterion and, in the second step, regions with similar intensities are grouped once more using a criterion homogeneity. These criteria influence the division of regions and subsequent growth of regions [Zuc76].

In [Dua+06] the authors use an undirected weighted graph to describe an over-segmented image. Each node represents a region in the graph, and edge weights measure the dissimilarity between regions according to their intensities. The authors use a heuristic to obtain the graph.

Freedman and Daniel [Fre10] improve the Random Walker method for segmenting. Their new algorithm proposes a new way to calculate edge weights and use a color probability distribution. In addition, they replace the traditional graph with a new graph-based on average change over-segmentation.

Shi and Malik [SM00] propose a novel graph-based segmentation method. Focuses on the overall impression of an image instead of focusing on local features and their consistencies. The authors use the "Normalized cuts" criterion to measure total dissimilarity between different groups and similarity between individuals in a group.

Faris et al. [AF+13] propose a region segmentation system using PSO. The SRG algorithm starts with a seed pixel and compares it with neighboring pixels according to attributes such as intensity or texture. In each iteration, it is joining them when they are sufficiently similar. This testing and ordering process is repeated for the following neighbors. In this article, the authors use Particle Swarm Optimization (PSO) to automate the selection of the initial seed pixel and the boundary value that measures the difference between the pixel and its neighbors.

In [BLM95] the authors use a genetic algorithm to find the optimal combination of parameters that maximize a segmentation quality criterion. The authors use the Phoenix [Law+82] segmentation algorithm, which is a recursive split segmentation technique. The authors argue that the algorithm can work with variations in the image that affect the quality of segmentation because the objective function they use adapts to these variations.

1.4.5 Techniques based on neural networks

On the other hand, genetic algorithms have not been used as support. The GA has been combined with neural networks. In most cases, genetic algorithms are used to optimize the parameters of the neural network cite GomezFlores2020.

Gomez and Flores [GFHL20] use a pulse-coupled neural network (PCNN) for image segmentation. However, the performance of PCNN is affected by hyperparameters. To face this problem, the authors use a differential evolution algorithm. In their experiments, the authors use the Clusters validity index as a fitness function.

Other cases, such as [DO08] combine a neural network with a genetic algorithm for tissue segmentation in ultrasound images, where the algorithms genetics automatically determines the topology and the number of nodes in the neural network during training. It is also the case of [XB13], in which a new algorithm for dermoscopy image segmentation is proposed using a combination of a self-generated neural network and the genetic

algorithm. A genetic algorithm is used to select the optimal samples as seeds, and these are taken as initial neuronal trees, and thus a self-generated neuronal forest is generated by training the rest of the samples.

There are semi-automatic segmentation processes as described in [Jan+13] an expert is needed to delimit the area to identify the spongy bone tissue that you want to segment in computed tomography. Subsequently, the genetic algorithm simultaneously searches for the correct graphic filters, precise parameters for these filters, and the correct order of application of the filters.

The Figure(1.1), shows the classification of state of the art above mentioned.

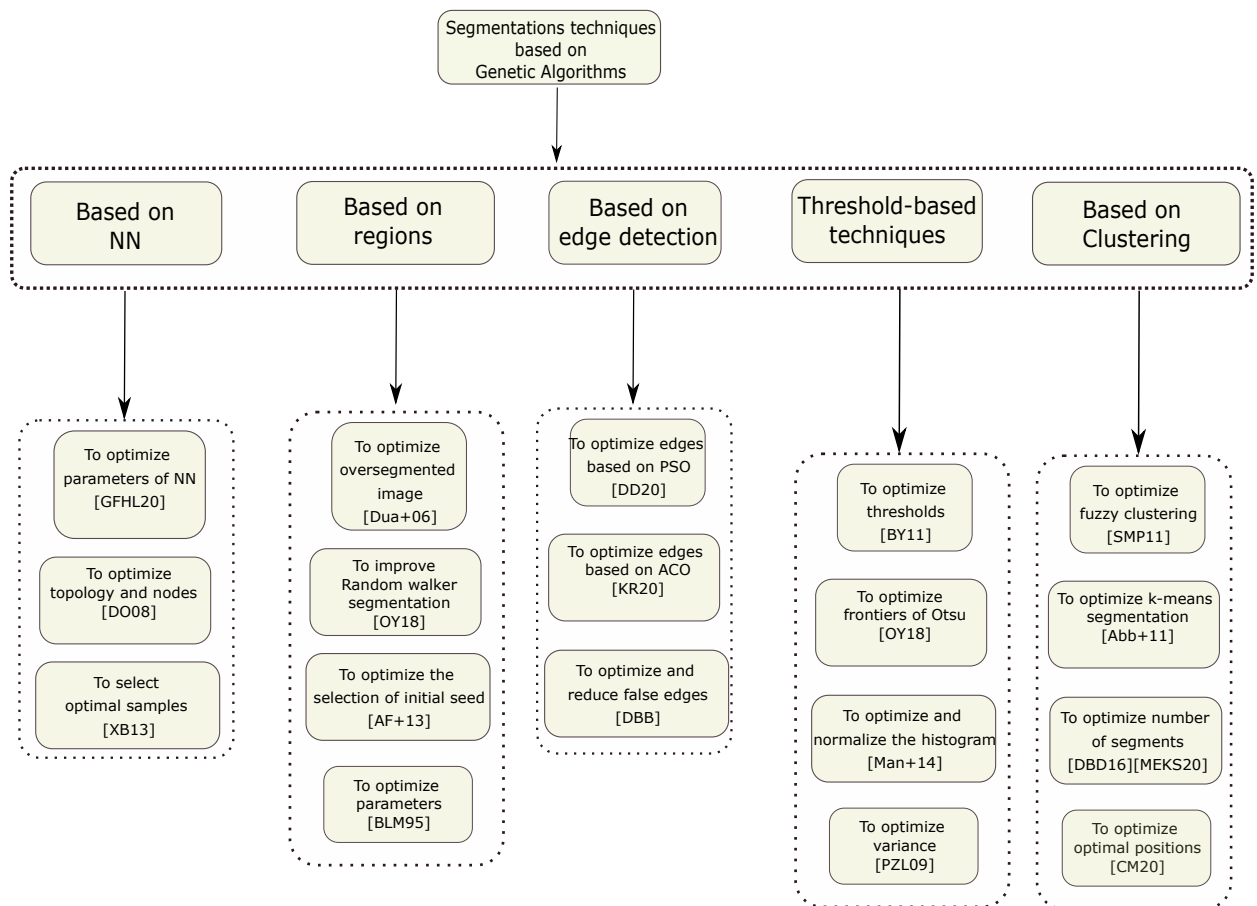


FIGURE 1.1: Papers in state of the art

Some articles discuss genetic algorithms to improve or combine them with a single segmentation technique. Others use some morphological operators, but in the same way, apply to the result of a single technique or combination. Some proposals are not entirely automatic and require the intervention of an expert in the area, and other

views previously run the risk of falling into a local optimum and resulting in poor segmentation.

1.5 Thesis organization

The rest of the Thesis is organized as follows:

Chapter 2 contains an overview of segmentation techniques and their classification, the performance metrics that will evaluate our work, and the theoretical framework of a genetic algorithm.

Chapter 3 shows the proposed methodology of this work, the processes and the tools used. A genetic algorithm is detailed to be applied in the solution of the segmentation problems.

Chapter 4 contains the experimental results of each tested data set, using performance metrics and comparing effectiveness with segmentation techniques in state of the art.

Finally, Chapter 5 is composed of the summary of the complete work, where some conclusions and ideas for later work are also discussed.

Chapter 2

Preliminaries

2.1 Segmentation techniques

Digital image processing (DIP) comprises many methods and techniques that are used to extract information from an image. One of the essential steps in DIP is image segmentation. Segmentation allows that an image to be broken down into its constituent parts, that is, the objects that are of interest and the background. It is based on specific local characteristics that allow distinguishing one object from another, simplifying the representation of an image.

A pixel represents the element of an image (picture element for its acronym in English). Each pixel gives us the necessary information about its gray level, color, texture, shape, etc. Each intensity of a pixel is represented by a simple number between 0 and 255. Segmentation is the procedure of assigning a value to each pixel that makes up an image, forming groups of pixels with the same value. In other words, segmentation is a multiclass classification where each class has specific characteristics or properties of the image to be examined. [ZA15] [Kha14].

Segmentation is based on the following principles:

1. **Similarity:** Each of the pixels of an element has similar values for some specific property. In this principle, the regularity in the gray level values is studied.
2. **Discontinuity:** The objects and, more specifically, the regions that differentiate them stand out from the environment and therefore have defined edges.
3. **Connectivity:** the pixels belonging to the same object or region have to be contiguous. That is, they tend to group constituting homogeneous regions.

Segmentation can be represented in two ways: semantic segmentation and instance segmentation.

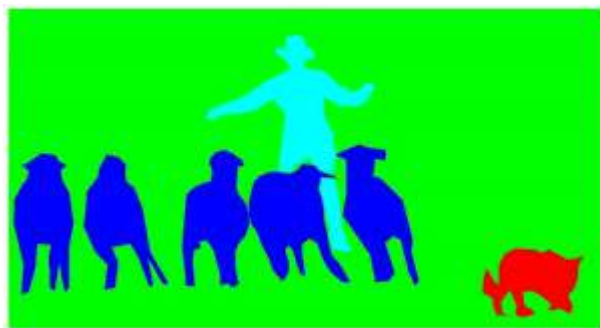


FIGURE 2.1: Semantic segmentation [Taken from <https://informatica.blogs.uoc.edu/la-segmentacion-semantic-y-sus-benchmarks/>]



FIGURE 2.2: Segmentation of instances [Taken from <https://informatica.blogs.uoc.edu/la-segmentacion-semantic-y-sus-benchmarks/>]

Semantic segmentation (Figure 2.1) permits the ensemble of several objects in the same group, which is the main definition of segmentation. In contrast, the segmentation of instances (Figure 2.2) will classify various groups with the same characteristics as different objects. On the other hand, panoptic segmentation is the union of semantic segmentation and instance segmentation.

The result of a panoptic segmentation is a new model that will have the peculiarities of the two previous models, the pixel label (semantic segmentation) and the prediction for each pixel instance (instance segmentation).

Image segmentation techniques are used to partition an image into several segments. These parts are used to locate the objects and their borders in the image. Effective segmentation is performed when the objects in the image are separated in an acceptable computational time and perfectly describe the object's boundaries. In general, the segmentation methods can be divided into two parts: pixel-based segmentation methods and texel-based segmentation methods, the most important are the first. Over the

last decades, various segmentation techniques have been developed. The Figure (2.3) describes the most elementary segmentation techniques.

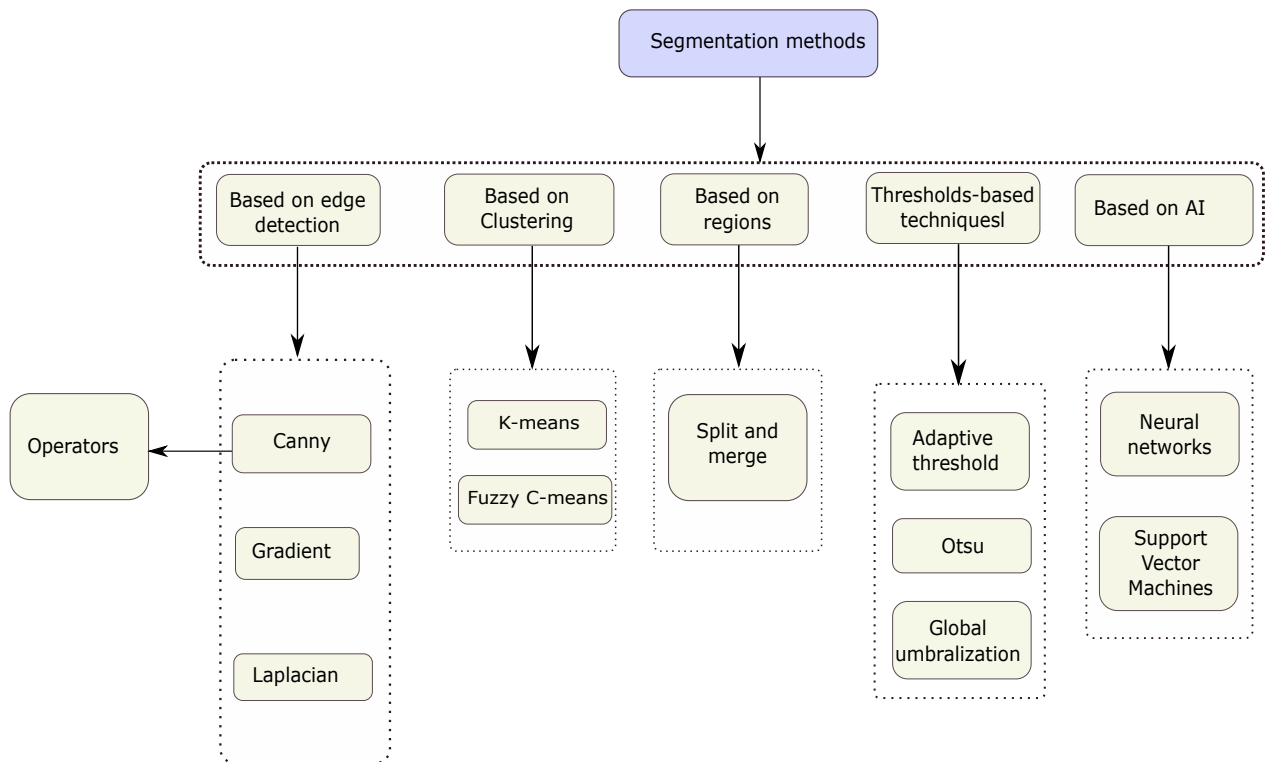


FIGURE 2.3: Elemental segmentation methods

In recent years there has been a great advance in the development of image segmentation algorithms. The numerous techniques developed can be classified according to the following categories:

1. Threshold-based segmentation techniques
2. Clustering-based techniques
3. Techniques based on edge detection
4. Region-based techniques
5. Techniques based on neural networks

2.1.1 Threshold-based segmentation techniques

Global Threshold

Thresholding is defined as a rapid segmentation technique with a low computational cost. Obtaining the threshold is considered under various criteria since it can be managed by the same user, where a threshold point is established or obtained automatically.

Automatically threshold is obtained in two ways, finding the threshold with the sum of the pixel with the highest intensity with the pixel with the lowest intensity and dividing it by two, as shown in Equation (2.1). The other way is by normalizing the histogram of an image I of size $M \times N$; this technique is simple because when obtaining the normalized histogram, which as is well known is the sum of the probabilities of existence of each pixel, does not consider spatial information but only the gray distribution in the image, Equation (2.2) shows how to obtain the normalized histogram.

$$th = \frac{\max(I) - \min(I)}{2} \quad (2.1)$$

$$th = \frac{\sum_{i=1}^m \sum_{j=1}^n f(i, j)}{M \times N} \quad (2.2)$$

Thresholding allows converting an image of gray levels into a binary image using Equation (2.3), where if it meets the condition of thresholding, the output will be 1, in otherwise 0.

$$G(x, y) = \begin{cases} 1, f(x, y) > th \\ 0, f(x, y) < th \end{cases} \quad (2.3)$$

where th Returns the value of the assigned threshold

The reason for using a histogram is the binomial distribution that is handled to implement a threshold that allows the visualization of various characteristics of the same image, thus avoiding the loss of information that could become important.

Adaptive threshold

Like the global threshold, the adaptive threshold is based on a threshold to obtain those necessary pixels to get some characteristic in common, and both return a binarized image. The difference between both is the algorithm for obtaining from the threshold. The algorithm here shows how to obtain the threshold by this method.

1. It start by selecting an initial border $th(i)$, where $i = 1$. The initial boundary can be any. In our case, $th(1) = 127$ was set. This threshold can be obtained in the same way with Equation (2.1).
2. The pixels are grouped based on th , thus creating two groups where the first group is for all those pixels with intensities greater than th and the second for pixels with intensities less than th The Equation (2.4) and Equation (2.5) show the pixel grouping.

$$G1 = \{0, \dots, th\} \quad (2.4)$$

$$G2 = \{th + 1, \dots, L\} \quad (2.5)$$

where $L =$ Maximum intensity present in the image

3. The average of each group is obtained based on whether they are greater than th or less than it. Equation (2.6) and Equation (2.7) show how to obtain the mean of each group.

$$\mu_1 = \sum_{i=0}^{th} \frac{G1(i)}{th} \quad (2.6)$$

$$\mu_2 = \sum_{i=th+1}^L \frac{G1(i)}{L} \quad (2.7)$$

4. Once the mean for both groups has been obtained, the border $th(i)$ is updated by using Equation (2.8).

$$th(i) = \frac{\mu_1 + \mu_2}{2} \quad (2.8)$$

5. Finally, this process is carried out until $th(i)$ is not modified for the result of two previous iterations. Otherwise, all the steps are repeated. Once the threshold is obtained condition shown in Equation (2.3) is used.

Otsu method

The Otsu method, named after Nobuyuki Otsu, developed in 1979, uses statistical techniques to solve the problem. Specifically, the variance is used, which measures the dispersion of values in gray levels.

Otsu's method calculates the threshold value so that the dispersion within each segment is as small as possible but at the same time as high as possible between different segments. The quotient between both variances calculates it. The threshold value is sought for which this quotient is maximum.

The Otsu method chooses the optimal threshold by maximizing the between-class variance through an exhaustive search. While there are different methods of finding a threshold, most of them do not work well with real-world images due to noise, flat histograms, or inadequate lighting. In contrast, Otsu's method was one of the best threshold selection methods for real-world images. However, this method uses an exhaustive search to evaluate the criteria for maximizing the variance between classes. As the number of classes in an image increases, the Otsu method takes more time to select a suitable multilevel threshold. The variance is obtained between two different classes of grouped intensities that are modified in each iteration to determine the threshold efficiently.

This modification of the method will considerably reduce the calculation time. The importance of the Otsu method is that it is automatic. That is, it does not require human supervision or prior information on the image before processing.

The Otsu segmentation algorithm is explained step by step below.

1. The segmentation of images by the Otsu method starts by normalizing the histogram of the image. It is achieved by applying Equation (2.9) on the image.

$$P_i = \frac{n_i}{N}; P_i > 0; \sum_{i=1}^L P_i = 1 \quad (2.9)$$

Where p_i = Probability of each pixel in the image. n_i = Total number of holdings of a specific pixel in the entire picture. N = Total number of pixels in the entire image.

2. Once the above equation is applied, it is assumed that the pixels are divided into two classes: C_0 with gray levels up to a threshold K , and C_1 with gray levels starting from $K + 1$ up to the highest intensity L . The expressions (2.10) and (2.11) show the grouping for each class.

$$C_0 \subset \{1, 2, \dots, k\} \quad (2.10)$$

$$C_1 \subset \{k + 1, \dots, L\} \quad (2.11)$$

3. The distribution of the probabilities of occurrence of the intensities that exist in each class is obtained. In order to obtain them, the equation (2.12) is applied in the case of those belonging to the class C_0 and the Equation 2.13 for class C_1 .

$$w_0 = P_r(C_0) = \sum_{i=1}^k P_i = w(k) \quad (2.12)$$

$$w_1 = P_r(C_1) = \sum_{i=k+1}^L P_i = 1 - w(k) \quad (2.13)$$

4. Then we proceed to obtain the respective means for each class. For class C_0 Equation (2.14) is used and for class C_1 Equation (2.15) is used.

$$\mu_0 = \sum_{i=1}^k i P_r(i | C_0) = \sum_{i=1}^k i \frac{P_i}{w_0} = \frac{\mu_k}{w(k)} \quad (2.14)$$

$$\mu_1 = \sum_{i=k+1}^L i P_r(i | C_1) = \sum_{i=k+1}^L i \frac{P_i}{w_1} = \frac{\mu_T}{1 - w(k)} \quad (2.15)$$

5. The variances are obtained, for the class C_0 is used the Equation (2.16) and for the class C_1 is used the Equation (2.17).

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 P_i(i | C_0) = \sum_{i=1}^k \frac{(i - \mu_0)^2 P_i}{w_0} \quad (2.16)$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 P_i(i | C_1) = \sum_{i=1}^L \frac{(i - \mu_1)^2 P_i}{w_1} \quad (2.17)$$

6. Finally the Equation (ref formula23) is calculated for each iteration. The smallest resulting value is taken as the threshold once the final result is obtained for each intensity level.

$$\sigma_w^2 = w_0\sigma_0^2 + w_1\sigma_1^2 \quad (2.18)$$

2.1.2 Techniques based on edge detection

Based on finding the border of a region, or the border that divides two regions. Edge-based segmentation, an image is segmented by identifying the edges or borders of objects. Boundaries are located by changes in the intensity image and are also used to identify discontinuities in the image. The purpose of edge detection is to significantly reduce the amount of data to be processed in the image while preserving the structural properties of the image [WJW09] [WO10].

Border-based techniques are used to convert grayscale images to binary images. This technique can use binary and multi-level segmentation based on the number of segments in the image. Binary segmentation divides the image into two different regions using a threshold value. All intensities in the image greater than that value are classified as object pixels, while all pixel intensities less than that boundary value are classified as the background of the image [Aro+08].

The Sobel techniques and the Prewitt operator have been discussed as first-order derivatives. These techniques detect the edges by finding the maximum and minimum of the first derivative of the image. The dissimilarity in pixels is the basic principle of techniques based on edge detectors.

Border detection

The edge in an image is a considerable difference in intensity between two pixels, a significant variation in the similarity of the value of these pixels. Therefore, edge detection techniques find color discontinuities or abrupt changes in brightness in some areas of the image.

Successful edge-based segmentation is due to three key steps: Detecting edges, removing irrelevant edges, and connecting or group pixels. [GL+18] [CS10].

The disadvantages of these segmentation techniques are:

- Noise sensitive technique.
- The technique is not appropriate if the edge of the image is wrong definite. Performance is not optimal.
- There are considerable errors in the segmentation if any many edges in the image.

Edge detection operators The Sobel operator is a classic first-order edge detection operator used to approximate the absolute magnitude of the gradient of each pixel using convolution matrices or a 3×3 filter pair. These filters estimate the gradients in the horizontal and vertical directions and the magnitudes of the slope.

Sobel operators are less sensitive to noise (as it uses larger convolution kernels) and can smooth the input image to a greater extent. But it produces thicker edges. The Prewitt gradient edge detector works similarly to the Sobel operator but uses a slightly different kernel. The Prewitt operator is an edge detector, based on small filters, separable and valued in integers with the horizontal and vertical direction. The advantages of the Prewitt operator are a very low computational complexity because the calculations are performed with only 1's.

The Canny Edge Detector is an optimal edge detection operator used to detect a wide range of edges in images. The input to a Canny operator is a grayscale image, which produces an output image showing the positions of the intensity discontinuities in a row.

Canny Edge Detection involves the following steps. A Gaussian convolution kernel smoothes the image. After that, the edge directions are estimated from the slope of the intensity surface of the smoothed image. The base algorithm establishes thresholds with local estimates of image noise. Two thresholds are used to deal with the problem of stripes. This algorithm obtains as many real edges in the image as possible, reduces false edges and generates good localization [Can86].

Gradient method

The primary definition is that the magnitude of the scene intensity gradient is much greater than at points spaced from the edge. Thus, the set of elements belonging to the edge in which the magnitude of the intensity gradient is high enough will represent the edge of the object extracted. Unfortunately, in practice, this set contains elements that do not belong to the structure and/or components that truly do belong are missing. There are several optimization strategies to overcome this problem. One possible solution is to assign a cost to each element on edge and find an edge whose total cost is the lowest of all possible ones. This method requires high computational power. For this reason, one usually chooses to make assumptions derived from the nature of the edges and the observation. Another solution is to start with an element of any border and add, of all the possible neighbours, the one that is more qualified each time.

Laplacian

For the gradient method, a single derivative is used to differentiate the image. In the Laplacian method, a second derivative is used to estimate better the edge's position, where the second derivative crosses zero.

The Laplacian of an image $I(x, y)$ is a derivative of order two defined by:

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.19)$$

In general, the negative value of the Laplacian is usually taken. As in the case of the gradient, it is implemented digitally in various ways.

Since the Laplacian is a derivation operator, the sum of the coefficients must be zero. Furthermore, the coefficient associated with the central pixel must be positive and the other coefficients negative or zeros. In the Figure (2.4) we can see a filter for the Laplacian calculation.

0	-1	0
-1	4	-1
0	-1	0

FIGURE 2.4: Laplacian convolutional filter

In this case, the expression to determine the Laplacian is given by (2.20).

$$4z_5 - (z_2 + z_4 + z_6 + z_8) \quad (2.20)$$

Although the Laplacian responds to transitions in image intensity, it is rarely used in practice. Because it is a second derivative operator, it is overly sensitive to the presence of noise.

2.1.3 Clustering-based methods

Based on finding regions or subregions where there is a constant in that part of the image. Pixel similarity is the basic principle of clustering-based segmentation.

In these methods, pixels are grouped into larger regions according to their similarity according to predefined similarity criteria and considering spatial adjacency relationships between pixels. Simple examples of similarity criteria could be [GL+18]:

- The absolute intensity difference between a candidate pixel and the starting pixel must be within a specified range
- The absolute intensity difference between a candidate pixel and the average operating intensity of the crop region must be within a specified range
- The difference between the standard deviation in intensity over a specific local neighborhood of the candidate pixel and that of a local neighborhood of the candidate pixel must (or must not) exceed a certain threshold; this is a fundamental criterion of roughness/smoothness.

K-means

K-means is a grouping method, which aims to partition a set of n elements into k groups in which each element belongs to the group closest to the calculated mean. It is generally a method used in data mining to calculate the distribution of a set of elements that you have, based on the distances of each component to the number of centroids.

The algorithm that will be used is based on the steps shown here.

1. K-points are chosen as centroids from our image to group the data.
 - item The distance of all the elements is calculated from them the number of centroids. Generally, this distance is calculated using Euclidean distances and adding the point to the centroid with the shortest distance.
 - item The mean for each point is calculated, and the position of the new centroid is obtained.
 - item The before process is carried out until there are no changes in the centers.

2.1.4 Region-Based Methods

Region-based methods are methods that usually start with a pixel and add pixels according to a similarity parameter. Thakare and Parandwadi [Pun11] argue that the crescent region technique is one of the most popular segmentation methods. This method starts with a pixel and adds pixels based on their similarity, forming a region. The whole

process is repeated until all connected pixels belonging to the same region are easily found. However, a disadvantage of this method is that noise causes poor performance and is computationally very expensive.

Split and Merge

As the name implies, the first step in this method is to split our main image or source based on a four quadrants data structure. The image is divided into four quadrants, and each quadrant can be divided into four more. This process is repeated repeatedly until reaching the goal of having pixels of the same color in a single quadrant.

The second step is to merge the regions that are adjacent and similar. This last process is done until it is not possible to perform any more merges [ZA15] [Nin+10].

Normalized cuts: These normalized slices are based on graph theory and have optimal division by reducing the number of regions. Each pixel is a vertex in a graph; the edges link adjacent pixels. Edge weights are assigned based on distance, gray level, similarity, color, or textures, etc. between two corresponding pixels [ZA15] [VKR08] [PZY10].

2.1.5 AI-Based Methods

Methods based on artificial intelligence (AI for its acronym) are a series of techniques performed by machines that emulate characteristics or unique capabilities of the human intellect, maximizing performance through learning from experience and adjusting to new contributions.

Neural networks

Artificial neural networks (RNA for its acronym) are a model where they simulate the behavior of biological neural networks that exist in the human brain; their objective is to solve problems through the flow of information and the adjustment of values that occur in the links of these neurons.

In general, an artificial neural network comprises three layers: input, hidden layer (s) and output.

One of the advantages of this model is the high degree of parallelism that reduces the computational time for solving a problem. This model learns and corrects its error using a technique called backpropagation.

Support vector machines

Support vector machines (SVM) is a binary classification method applied to create an optimal hyperplane. The margin of separation between the two classes in the data is extended to the maximum and thus separate what is not linearly separable.

In the context of segmentation, it is used to classify pixels. Pixels with characteristics in common are treated as an object, and those pixels with a different value are treated as a background.

2.2 Color spaces

Color space is a specific organization or range of colors in an image, video, computer, device, or human sight. In other words, it is the combination of primary color pigments and adding a mapping function. The objective of a color model is to facilitate the specification of colors in a standardized and generally accepted way. In essence, a color model is the specification of a three-dimensional coordinate system and a subspace of this system where each color is represented by a single point in a three-dimensional coordinate system.

The importance of color is a robust descriptor to identify and extract an object from a scene. Although humans can distinguish thousands of colors, they can only distinguish two dozen in gray scale. Standard hardware-oriented models are RGB (red, green, blue) for monitors and camcorders; CMY (cyan, magenta, yellow) for printers and YIQ (where Y is brightness and I and Q are chromatic components), which is the standard for television [Gon02].

In the literature, the color spaces for color image processing are: RGB, HSV (hue, saturation, value), HSI (hue, saturation, intensity), CIE XYZ, NTSC, YUV, and YCbCr [GL+18].

2.2.1 RGB space color

In the RGB model, each color appears in its spectral components: red, green, and blue. This model is based on the Cartesian coordinate system. The color subspace of interest is the tetrahedron shown in the Image (2.5), in which the RGB values are at three vertices.

Black corresponds to the origin, and white is the furthest from the origin at the opposite vertex. In this model, the grayscale runs from black to white, and colors are

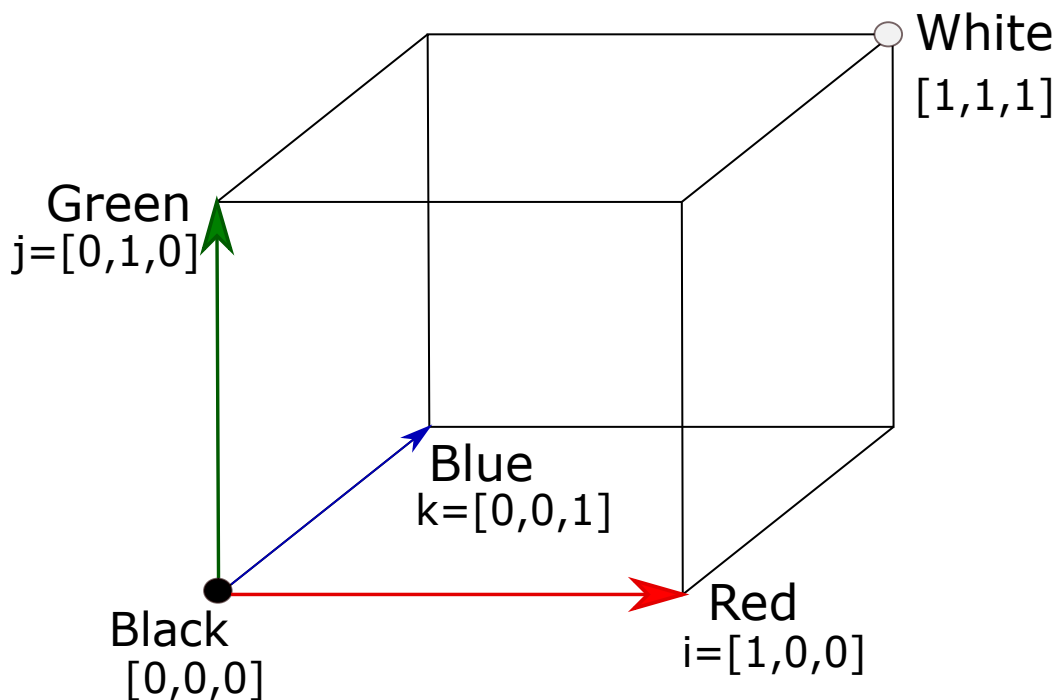


FIGURE 2.5: RGB space color model

points within the tetrahedron defined by the vectors from the origin. For convenience, vectors are normalized. All the values of R, G and B are in the range $[0,1]$. The images in this model are formed by combining each of the RGB primary colors in different proportions. RGB space is a simple model and one of the most used in Digital image processing.

2.2.2 HSV space color

In this space, color is represented by the hue (H), saturation (S), and value (V) components. Hue is the chromatic characteristic that describes a pure color: yellow, orange, red, etc. Saturation is a measure of how the hue is diluted in white light; The value is the intensity or brightness of the color.

The HSV space becomes a helpful tool to develop image processing algorithms based on some properties of human perception of color [\[Gon02\]](#) [\[Ito+06\]](#).

The pitch is in the range $[0, 2\pi] \subset R$; the saturation is in the real range $[0, 1]$, while the value is usually in the range $[0, 255] \subset R$. The HSV space has a shape where the tip is black, and it becomes lighter until it reaches white, which is the base. The radius

and the height of the cone represent the saturation and value components, respectively. The Figure (2.6) shows the HSV representation.

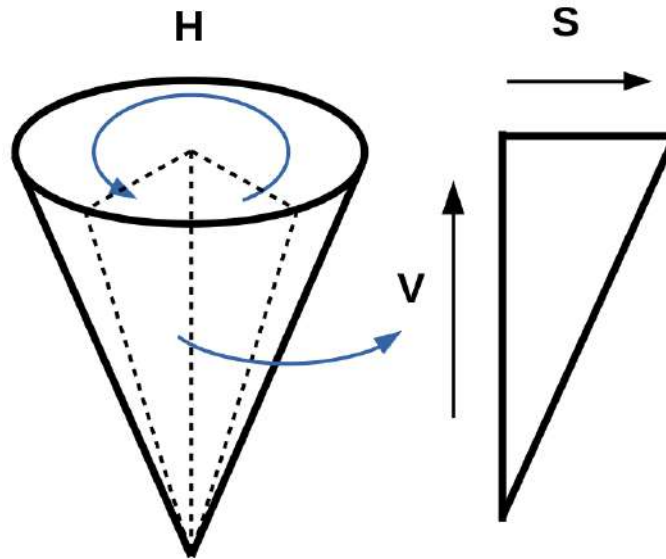


FIGURE 2.6: HSV spcae color model [Taken from https://www.researchgate.net/figure/Figura-23-Espacio-de-colores-HSV_fig2326551993]

In HSV space, the tone parameter is not defined, because these colors are considered singularities within this color space; because they do not have a specific chromaticity [GL+18].

2.2.3 HSI space color

Its acronyms correspond to H (Hue), S (Saturation), I (Intensity). The HSI model is useful for two characteristics:

- The intensity component (I), is decoupled from the color information contained in the image.
- The components of hue and saturation are closely related to the way humans perceive color.

Component ranges are the same as HSV space. That is, the pitch is in the range $[0, 2\pi] \subset \mathbb{R}$; saturation is in the real range $[0, 1]$, while intensity is in the range $[0, 255] \subset \mathbb{R}$. The

difference is that the HSI space has a double cone shape; therefore, its components are calculated differently.

2.3 Genetic algorithms

Genetic algorithms (GAs) start from using the theory of natural evolution developed by Charles Darwin. GA use natural selection to solve optimization problems [GH88] [Pap+05]. Optimization is done through the natural exchange of genetic material between parents. Descendants are formed from parental genes. The offspring's fitness is evaluated, and the individuals with the best aptitude have the best possibilities to reproduce. In the computer world, genetic material is replaced by bit strings and natural selection is replaced by the fitness function. The crossing of the parents is represented by crossing and mutation operations.

Elements of a genetic algorithm.

In GA, an initial population of individuals is required. Each individual is a candidate to be the solution to the problem addressed. The chromosome of each individual allows reaching a solution.

Each individual in the population is represented by a binary chain called the genotype of the individual analogous to the chromosome in the biological system.

Each genotype represents x points of the problem search space. Each point x is called a phenotype. The term gene is used to refer to the coding of a specific characteristic of the individual. Each gene can take different values that are called alleles. The term locus defines a specific position in the binary chain.

The Figure (2.7) shows the different expressions used in GA.

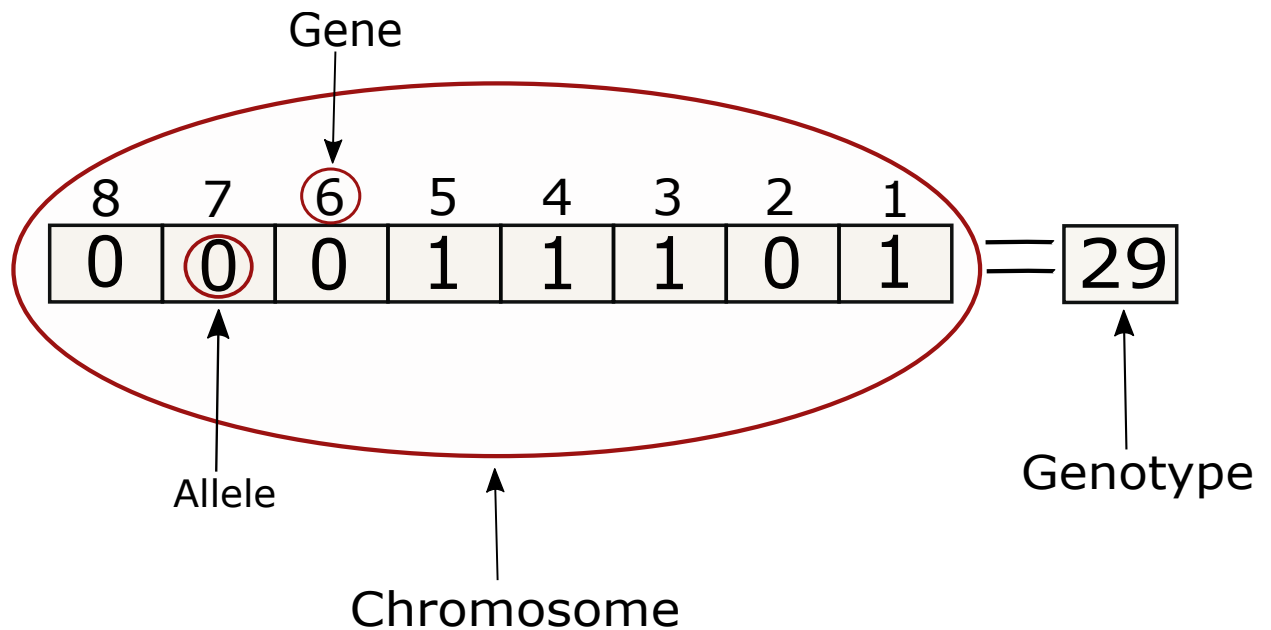


FIGURE 2.7: Graphic representation of expressions used in GA

2.3.1 Basic genetic Algorithm

Basic genetic Algorithm use the most straightforward possible operators in each of its stages.

The GA starts with a randomly generated population. A mathematical function is used to find the optimal value in a specific interval. The GA starts a cycle where the first step is a selection of individuals. This selection is made in such a way that the best-adapted individuals remain. The individuals to be crossed are chosen consecutively since the selection process is assumed to improve the fitness of the individuals efficiently.

A random mutation is applied, and the level of adaptation of the new generation of individuals is determined. An unemployment criterion is a maximum number of generations in which there was no improvement in skills.

The Algorithm shows the general scheme of a basic genetic algorithm. (1)

The parts of this algorithm are described in more detail below:

2.3.2 Initial population

The individuals in the initial population are usually strings of zeros and ones generated entirely randomly. Each gene is generated with a function that returns a zero or one

Input:	variables, size of chromosoma, ranges, etc .
Output:	Best solution

- 1: *Generate initial population*
- 2: *Compute fitness of initial population*
- 3: **WHILE** *stop condition is not met* **Do**
- 4: *Selection*
- 5: *Crossover*
- 6: *Mutation*
- 7: *Obtain a new generation of individuals*

END

Algorithm 1: Basic Genetic Algorithm

with equal probability. It is crucial to provide the population genetic algorithm with sufficient variety to explore all areas of the search space.

2.3.3 Selection

The basic idea of selection is to use a probability distribution, where the probability is directly proportional to the fitness function. The selection process must favor the number of copies of the most adapted individuals. The selection techniques can be classified into three groups: proportional selection, tournament selection, and uniform state selection.

Roulette Wheel Selection This method is the most used since the beginning of GAs. The algorithm presents the problem that the least fit individual can be selected more than once. However, its popularity is due to its simplicity. The roulette algorithm is as follows:

- Calculate the sum of expected values T .
- Repeat N times (N is the size of the population).
- Generate a random number r between 0.0 and T .
- Cycle through the individuals in the population adding the expected values until the sum is greater than or equal to r .
- The individual who exceeds the limit sum is selected.

2.3.4 Crossover

Crossover is a method of fusion on the genetic information of two individuals. This process provides a mechanism to inherit characteristics to their offspring where both parents intervene.

The simplest form of the crossover operator is the single point crossover, which consists of selecting a single position in the chain of both parents and exchanging the divided parts for that position. The Figure (2.8) shows an example of crossover operator.

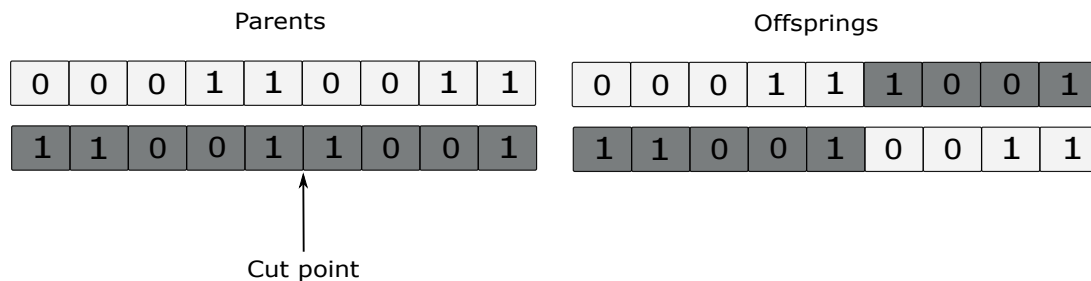


FIGURE 2.8: Crossover operator

2.3.5 Mutation

Mutation is a process where the genetic material is altered randomly due to errors in reproduction or deformation of genes. Unlike in human genetics, the probability in a genetic algorithm is higher. In fact, in a genetic algorithm, the mutation is a way to avoid falling into local minima.

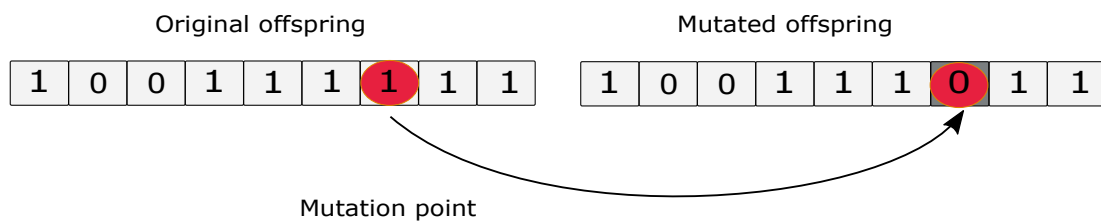


FIGURE 2.9: Mutation operator

The simplest form of mutation is to change the value of one of the positions in the chromosome. If the value is zero, it goes to one, and if it is one, it goes to zero. The Figure (2.9) shows an example:

2.3.6 Stop condition

It is necessary to specify the conditions in which the algorithm stops evolving and the best solution is found. The most straightforward condition used is when the algorithm detects that most of the population has converged to a similar form, lacking sufficient diversity to make it meaningful to continue the evolution.

Chapter 3

Metodology

In this chapter, the methodology used to optimize segmentation is described step by step. The detection of objects in a vision system has been important for computational advances and applications in our daily lives. Vision Techniques have been used in numerous applications such as detecting diseases in plant leaves, detecting different types of cancer, facial recognition for security cameras, and construction detection are some of the many applications of these vision systems where some of These examples of vision systems need to go through a segmentation process beforehand. Figure 3.1 shows the complete process of a vision system and shows how segmentation is a single step within that process.



FIGURE 3.1: Component of a vision system

To segment images, several segmentation and pre-processing techniques can be used. Each of them, or a combination of these, will give us a certain quality in the segmentation. On the other hand, each image has specific characteristics, and each one of the images can be affected by changes in lighting or noise. Therefore, the segmentation quality will be different for each one.

Testing each segmentation and pre-processing technique and choosing the optimal combination is a challenging and expensive task. An expert often does this work. However, small changes in lighting or noise can significantly affect the quality of the segmentation. That is why it is essential to include a genetic algorithm that searches for the right combination of parameters and techniques to obtain the optimal segmentation.

This work includes some segmentation techniques combined with some pre-processing techniques and morphological operators using some color spaces. Figure 3.2 broadly shows what has been said above. This chapter describes the process carried out step by step.

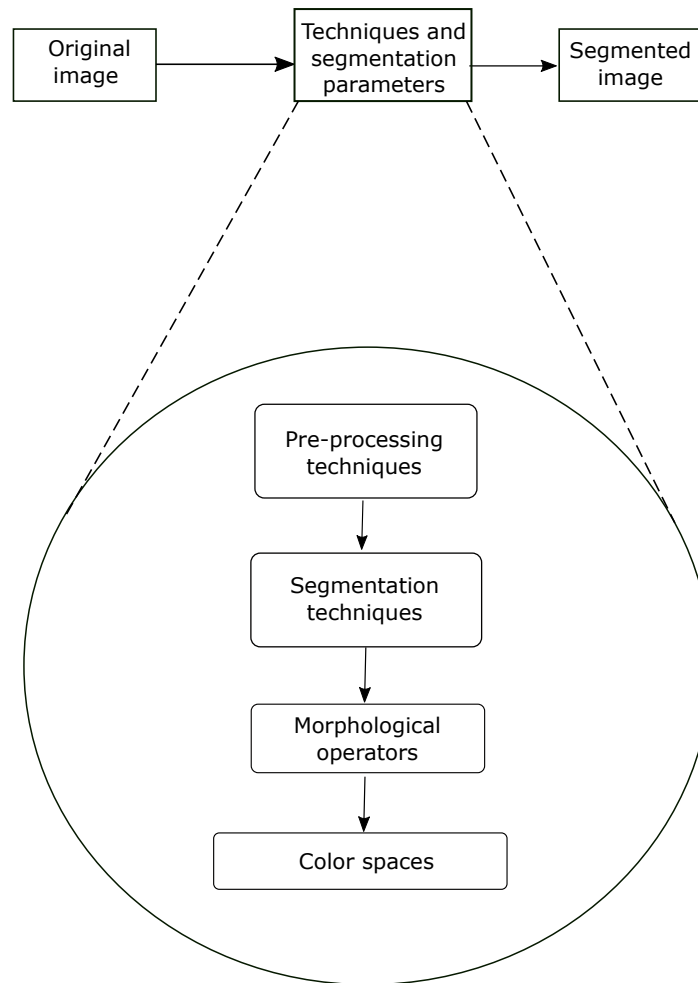


FIGURE 3.2: Techniques and segmentation parameters

The solution proposed within the methodology of Figure 3.2 begins by taking an original image, which we will call Image E, with this image, a set of techniques with associated parameters that were previously optimized using a genetic algorithm, and an Image S is obtained.

The genetic algorithm searches for the technique that obtains the best results from various preprocessing techniques, segmentation, morphological operators, and color spaces. Each combination of segmentation techniques with their respective parameters

is encoded and makes up a chromosome representing an individual. This individual is evaluated, or her aptitude quantified through some technique to evaluate the quality of the segmentation. Individuals with the best fitness have a greater probability of survival and a high probability that some of their genes (combination of techniques) will be passed on to the next generation. This process is carried out iteratively for several generations until a stop condition is met or optimal segmentation is obtained. The correct combination of technique and parameters has been found to achieve good segmentation in the data set. The following sections describe each of the steps of the proposed methodology in detail.

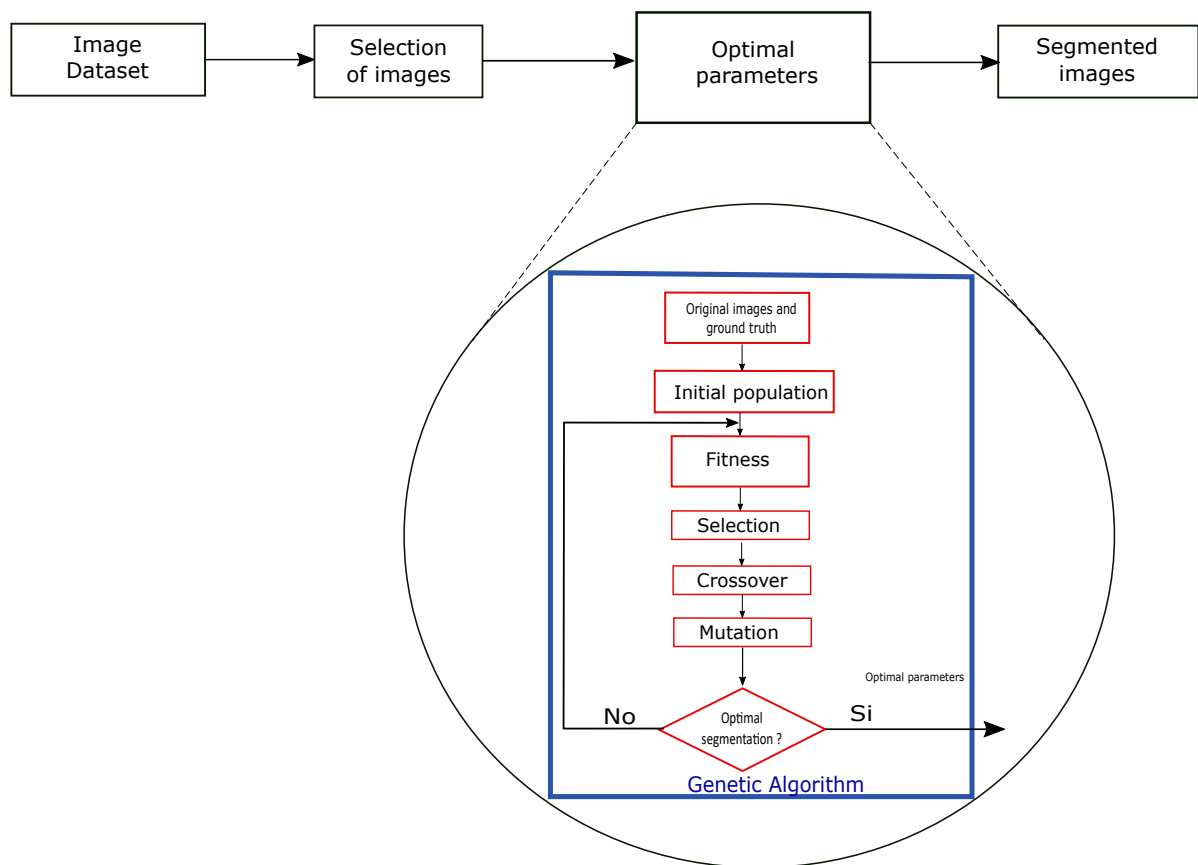


FIGURE 3.3: Proposed methodology

3.1 Datasets

The first step is the selection of the data set. The proposed method is designed to work on data sets with a constant background. There may be some minor variations, especially in lighting and noise. However, in general, the image sets have few variations. The figure shows several sample image sets, each row representing one set.

The first row belongs to an ICL plant leaf image set, which is a collection of leaves from the University of Hefei [Hu+12]. The data set is easy to segment as it does not have significant changes in lighting, and the background is entirely white. Only a few minor changes to the surface of the leaf could affect the segmentation of the leaf. However, it seems to be very easy to find the segmentation techniques and parameters for optimal segmentation.



FIGURE 3.4: Datasets used in the experimental results

The second row belongs to a set of images of the Mexican sign language. The data set belongs to a collection of images and videos of words in the Mexican sign language [EC+21]. In this case, even when the background is black, and the people are dressed in black, it is possible to see different changes in lighting in the images. These lighting changes significantly affect the segmentation of the image. Finding a combination of techniques and parameters that perfectly segment the set of images, in this case, is a more complex task.

Finally, the third row belongs to the Plantvillage [MHS16] data set; it contains images of plants with different affectations or diseases. Although the data set seems to have the

same background, it changes in intensity and tonality. In addition, the images present various shadows that affect the quality of the segmentation. In addition to the above, the gloss on the leaves can also cause poor segmentation.

3.2 Image selection

The genetic algorithm must calculate the suitability of a combination of techniques and parameters to get the best combination. For this, we can calculate the aptitude by evaluating how a combination of parameters behaves with an image.

It could give erroneous results, as it is better to assess fitness on several images rather than on a single image. In this thesis, several images are selected that together contain the variety of characteristics of the whole set; they are segmented by hand and are labeled as optimal segmentation.

The segmentations obtained are compared with the optimal ones, and the more similar they are to the optimal segmentation, the greater the aptitude of the combination of techniques and parameters. The subset of images that we will segment by hand and use with the genetic algorithm to obtain the optimal parameters is selected manually. However, it is possible to use clustering algorithms to obtain the most representative images of the images automatically.

3.2.1 Sefexa Image Segmentation Tool

We use this tool to segment the subset of images with varied characteristics. We refer to varied characteristics to images with high lighting or low lighting, images that present shadows, etc. It allows the genetic algorithm to learn to segment over a broad spectrum of images, considerably reducing the error when segmenting over the whole data set. It allows us to obtain the necessary parameters on images with various characteristics, as can be seen in Figure 3.4. The third row shows images with different levels of illumination, including images with shadows and images with a target image with different tonality. These images with different characteristics should be selected to improve the overall segmentation quality.

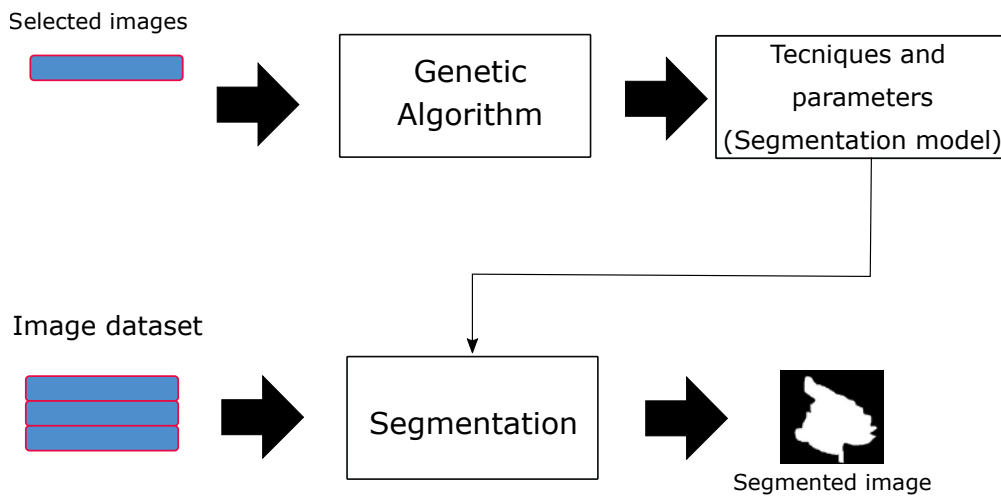


FIGURE 3.5: Selection of the set of images and obtaining of segmentation parameters

Figure (3.6) shows the Sefexa software interface with an example of an image of a sheet. Segmentation is straightforward and intuitive. It allows us to manually segment images with shadows or small objects that we do not want to be part of the segmented region.

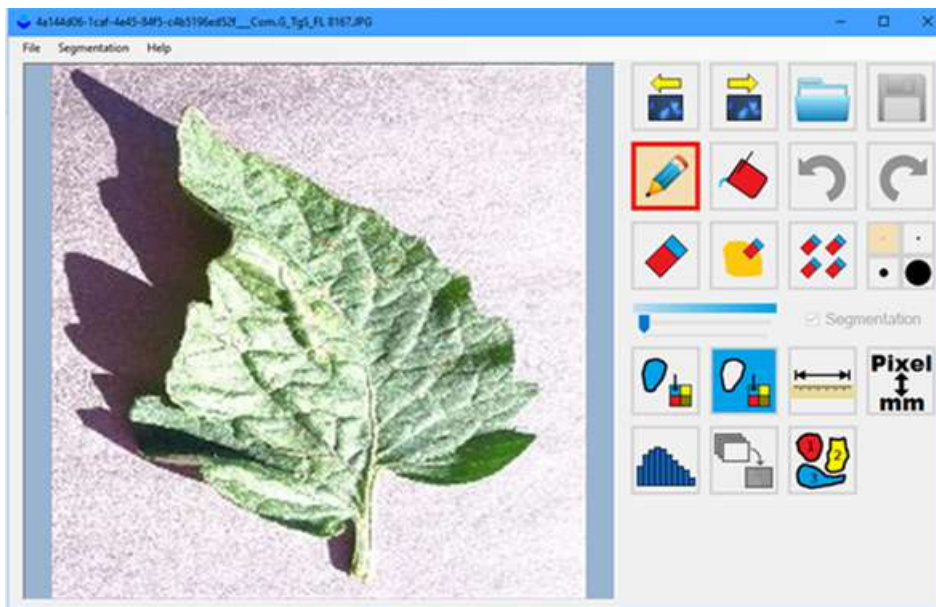


FIGURE 3.6: Sefexa software interface

In the Figure (3.7) you can see the final segmentation of an image that presents some shadows and that we do not want to be part of the target region. In manual segmentation, it is straightforward to segment an image. However, it is almost impossible to segment each image manually when we work on huge data sets. In these cases, an automatic or semi-automatic segmentation system is essential, as is the case proposed.

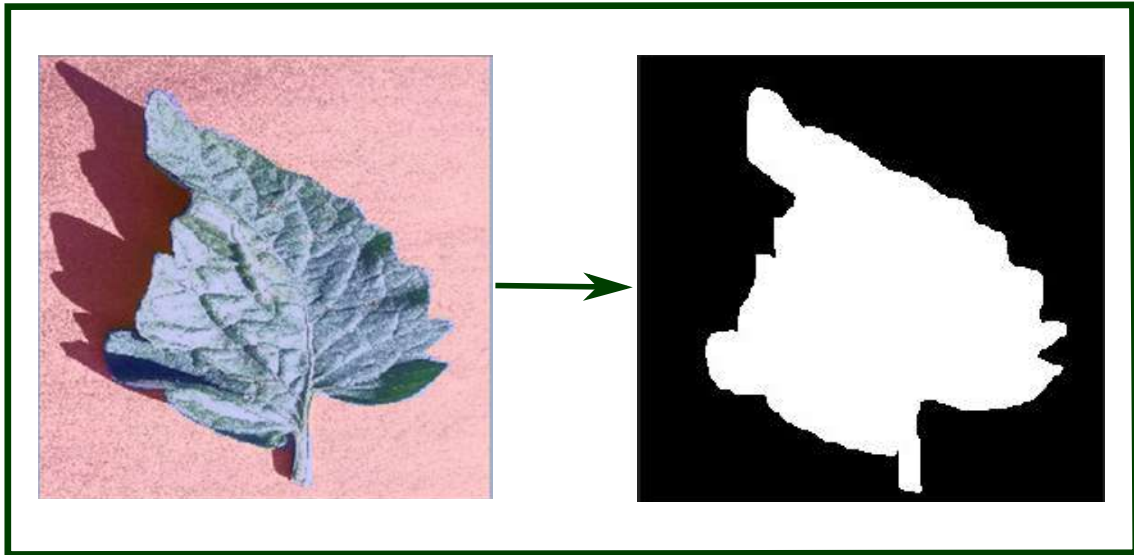


FIGURE 3.7: Segmentation by sefexa software

3.3 Selection of techniques and parameters

This is the core part of the proposed methodology. The segmentation problem can be seen as an optimization problem, where $\Omega = \{S^1, \dots, S^n\}$ is the set of candidate segmentation parameters, that is $S = \{p_1, p_2, \dots, p_t\}$, therefore, optimal segmentation $S^* \in \Omega$ must satisfy the following:

$$\forall S \in \Omega : f(S^*) < f(S) \quad (3.1)$$

where $f(\cdot)$ represents the fitness function, which measures the obtained and optimal segmentation relationship. A good similarity between the two segmentations will give us a suitable fitness function.

Since genetic algorithms help us optimize functions based on a fitness function, in this thesis, a genetic algorithm is proposed that obtains the appropriate segmentation techniques and parameters. Genetic algorithms are one of the most used heuristics to

solve real optimization problems. This algorithm is based on the natural evolution of the species. Genetic algorithms search within a set of solutions for the best solution using an initial set of individuals representing a set of solutions. These solutions are recombined and mutated based on a selection criterion generally based on the individual's fitness, producing new individuals with the best fitness in each generation. This process is repeated until a solution is reached or a stop criterion is met. The genetic algorithm involves the following steps, which are shown in Figure 3.8.

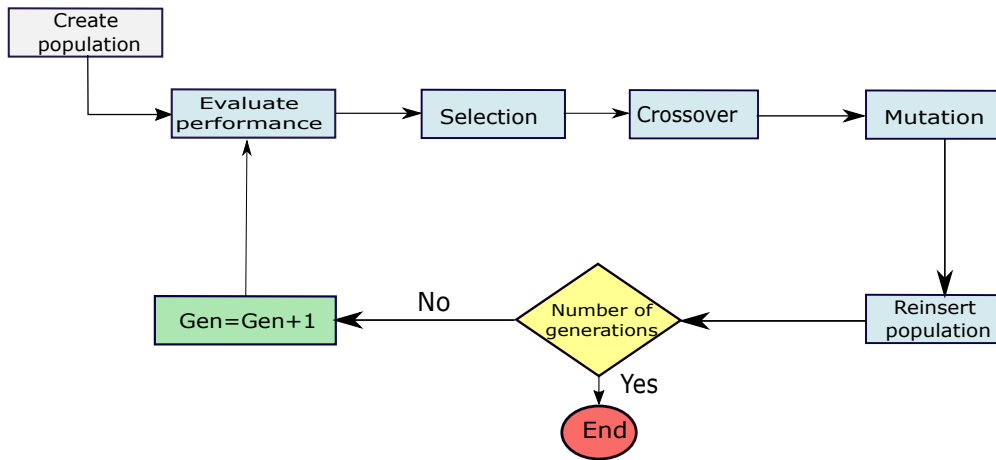


FIGURE 3.8: Genetic algorithm flow chart

3.3.1 Initial population and chromosome representation

Just as in biology, species evolve, and the strongest or fittest individuals survive. In a set of possible solutions to a problem, each of these solutions works as if they were an individual; therefore, we have a population of possible solutions where we seek to find the individuals with the most aptitude through several generations.

The population is obtained randomly, where the number of alleles of the chromosome depends on the number of variables. A population with n individuals is defined by $S^g = \{\mathbf{s}_1^g, \mathbf{s}_2^g, \dots, \mathbf{s}_n^g\}$, where g represents the current generation. A vector represents each individual with dimension d which is defined by $\mathbf{s}_i^g = [s_{i,1}^g, s_{i,2}^g, \dots, s_{i,d}^g]$

The variables in the search space that constitute the phenotypes are encoded in genotypes. The chromosome defines the genotype of an individual. On the other hand, the possible solution to the problem of that chromosome is the phenotype. The

genetic operators are applied to the genotype in each iteration. It allows improving the individuals.

In our case, two experiments were carried out. The first experiment with only nine variables and a second case was carried out because the results with nine variables were not very good. The results with 11 variables helped to improve the quality of the segmentation.

Case 1 (9 variables)

In the first, a random population of 120 individuals was created with nine variables that make up each one. It is, $\mathbf{S}^g = \{\mathbf{s}_1^g, \mathbf{s}_2^g, \dots, \mathbf{s}_{120}^g\}$ y $\mathbf{s}_i^g = [s_{i,1}^g, s_{i,2}^g, \dots, s_{i,9}^g]$

The 120 individuals are randomly initialized in the search space $[s_k^L, s_k^U]$ where k represents the k th chromosome variable, L and U represent the lower and upper bounds respectively for each variable. The dimensions used are described for each variable. The Image (3.9) shows our variables represented in a chromosomal chain, where each allele represents each variable of the algorithm.

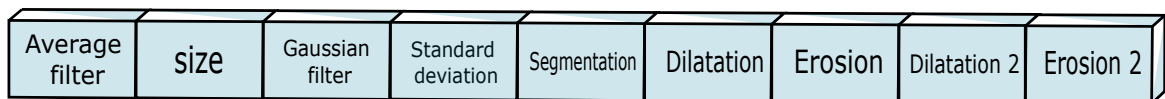


FIGURE 3.9: Chromosomal chain of nine variables

Donde:

- Average filter: It is a discrete variable that can take the value of 1 if an average filter is used or 0 if it is not used.
- Size: It is a discrete variable and refers to the size of the average filter. It can take the values of 3, 5, 7, and 9.
- Gaussian Filter: It is a discrete variable. The variable indicates that a Gaussian filter is used (1), or a Gaussian filter is not being applied (0).
- Standard Deviation: The standard deviation of the Gaussian filter is a continuous variable that can take values from 0 to 1.
- Segmentation: This variable represents the type of segmentation to be used. It can take four values, which are as follows.

- Clustering
 - PCA
 - Adaptive border
 - Otsu method
- Dilation: The dilation can take discrete values from 1 to 4, representing the circle's sizes to be dilated.
 - Erosion: Erosion as well as dilation, has 4 size values for this discrete variable.

The last two variables are repeated to improve the quality of the segmentation. The time of dilation and erosion used is different, and the combination of these generates a different final segmentation depending on the combination used.

Case 2 (11 variables)

After observing some tests with the nine-variable algorithm, it was decided to add two more variables to improve the segmentation quality. For the second case, a random population of 120 individuals was created with 11 variables that make up each one. That is, $\mathbf{S}^g = \{\mathbf{s}_1^g, \mathbf{s}_2^g, \dots, \mathbf{s}_{120}^g\}$ y $\mathbf{s}_i^g = [s_{i,1}^g, s_{i,2}^g, \dots, s_{i,11}^g]$

The 120 individuals are randomly initialized in the search space $[s_k^L, s_k^U]$ where k represents the k -th chromosome variable, L and U represent the lower and upper bounds respectively for each variable. The dimensions used are described for each variable.

The Image (reffig:cadena11variables) shows the complete chromosome.

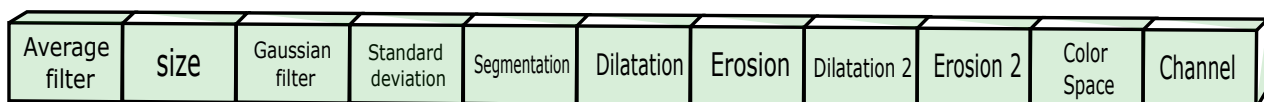


FIGURE 3.10: Chromosomal chain of eleven variables

where:

- Color space: Defined by the color space that can take 5 variables.
 - RGB

- NTSC
 - HSV
 - XYZ
 - YCBCR
- Canal: The channel has 4 settings which are described below.
 1. All the channels
 2. Firts channel
 3. Second channel
 4. Third channel

From the population, we evaluate each one of the individuals to know their fitness or, in other words, the quality of the segmentation obtained from the variables used.

Usually, each variable is decoded by introducing it in the Formula (3.2) to obtain its value. In our case, as can be seen in Figures 3.9 and 3.10 almost all variables are discrete; only one variable is continuous; this is the standard deviation used to the Gaussian filter.

The size of the chromosome section depends on the required search space. In our case, for the calculation of the standard deviation, a detailed search is unnecessary, so jumps of 0.1 were used, and since the standard deviation was limited with values between 0 and 1, only four alleles were needed to encode it.

$$x_i = x_i^l + \left(\frac{x_i^u - x_i^l}{2^\beta - 1} \right) (\gamma) \quad (3.2)$$

Where:

β = Number of alleles on the chromosome

x_i^l = Lower limit

x_i^u = Upper limit

γ = The binary number of the chromosome in decimal

The length of each binary substring depends on the size of the search space and the number of decimal places required for the decoded variable. If each decision variable is given by a string of length L and there are n variables, then the final length of the chromosome will be of size $L \times n$.

3.3.2 Fitness function assessment

The chromosome has each of the necessary variables to segment an image. The variables define the preprocessing, the segmentation technique, morphological operation, and the color space of the segmentation. Each combination of these variables results in a segmented image with a particular segmentation quality. The evaluation or quality of this segmentation with the variables that compose it is obtained by making a comparison with manual segmentation or optimal segmentation (*Ground truth*). *Ground truth* is a term commonly used to refer to the information provided by direct observation as opposed to information provided by inference.

To make this comparison between the direct information and the information obtained by inference, several performance metrics VOI, PRI, and GCE are used, which are described below.

Probabilistic Rand Index (PRI)

The probabilistic index defines the accuracy of segmentation from a statistical point of view. This index was introduced in [Ran71] to measure the similarity of clustering methods and modified by [UH05] to evaluate similarity in segmentation.

Once an image is segmented, this segmented image (S) has class 0 or class 1 pixel labels. The manually segmented image (G) also has class 0 and 1 labels. However, its labels were obtained manually according to prior knowledge.

The statistical probability index (PRI) counts the number of pairs of pixel labels equal in the segmented image S, and the manually segmented image G. PRI obtains the total average and limits it in the range [0,1]. A value of 0 means no consistency between the manually segmented image and that obtained through an algorithm. While a value of 1 means that the segmentation obtained with the algorithm is identical to that obtained manually.

The segmentation of an image can be described in the form of binary numbers $I(l_i^{s_k} = l_j^{s_k})$ on each pair of pixels (x_i, x_j) . The distribution of these numbers follows a Bernoulli distribution and gives us a random variable with an expected value denoted by p_{ij} . The probabilistic index of two segmentations can be defined as:

$$PR(S_{test}, S_k) = \frac{1}{N} \sum [I(l_i^{S_{test}} = l_i^{S_k})p_{ij} + I(l_i^{S_{test}} \neq l_i^{S_k})(1 - p_{ij})] \quad (3.3)$$

Where N is the number of pixels, S_k is the set of Ground truth segmentations, p_{ij} is the probability that the labels in the segmentation performed are equal to the actual segmentation. In practice, the mean is used to calculate p_{ij} . The probabilistic index takes values in the range $[0,1]$, where a score of zero indicates that the labeling of the test image is opposite to the actual segmentation, and a score of 1 indicates that the Ground truth segmented image and the test segmented image are the same for each pair of pixels.

A possible disadvantage of the PRI index is that it has a minimum dynamic range, and since images are often huge, this causes the PRI results to be very similar.

Global Consistency Error (GCE)

GCE allows calculating the degree of overlap of regions in the image. It was proposed in [Mar+] and [Mar03]. GCE was devised to quantify the quality of segmentation at different levels of detail. Let $R(S, p_i)$ be the set of pixels in segmentation S that contain the pixel p_i , the local refinement error is defined as:

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i)/R(S_2, p_i)|}{|R(S_1, p_i)|} \quad (3.4)$$

This error is not symmetric with respect to the compared segmentations and takes the value of zero when S_1 is a refinement of S_2 in pixel p_i . GCE is defined as

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i x_i E(S_2, S_1, p_i) \right\} \quad (3.5)$$

Variation of Information (VOI)

Information Variation (VoI) [Mei05] defines the distance between two segmentations as the average conditional entropy of one segmentation given the other, and therefore approximately measures the amount of randomness in a segmentation that cannot be explained by the other.

VOI measures the sum of the loss and gain of information between two segmented images (one real and one test); it measures the degree to which one class can explain the other. The VOI metric is non-negative, with lower values indicating high similarity. It is based on the relationship between a point and its class.

Information variance is a measure of the distance between two class (element) partitions.

A class with pixels X_1, X_2, \dots, X_k is represented by a random variable X with $X = \{1, \dots, K\}$. in such a way that $p_i = |X_i|/n_i \in X$ y $n = \sum_i X_i$ the variation of information between the classes X and Y is defined as

$$VI(X, Y) = H(X) + H(Y) - 2I(X, Y) \tag{3.6}$$

where $H(X)$ is the entropy of X and $I(X, Y)$ is the similar information between X and Y . $VI(x, y)$ measures the pixel allocation for a class of items X reduces the uncertainty about topic pixels in class Y .

3.3.3 Selection

In selection, the probability that an individual will be chosen and the probability that his seed will pass to the next generation is directly proportional to his fitness.

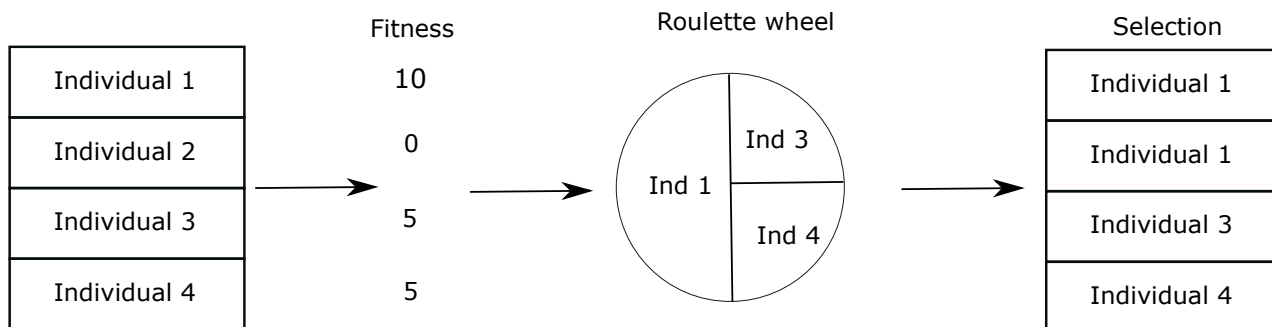


FIGURE 3.11: Representation of fitness percentage

In the experiments, we use the wheel roulette selection method. This technique assigns each individual a segment of a wheel. The magnitude of each individual’s wheel segment depends on the fitness obtained. The roulette wheel is rotated, and at the point where the wheel stops, it will indicate the individual to be selected to survive in the next generation. Figure 3.11 shows the selection process by the roulette method. The selection algorithm for the roulette method is described in the algorithm 2.

Input: Fitness of each chromosome f_i .
Output: Selected chromosomes

- 1: Calculate the total of fitness of the n individuals $\sum_{i=1}^n f_i$
 - 2: Generate a random number $r \in [0, 1]$
 - 3: **Get the value of** $s = r \times \sum_{i=1}^n f_i$
 - 4: If $s \leq \sum_{i=1}^k f_i$ then the k th chromosome is selected for the next generation
 - 5: Repeat steps 2 to 4 until the number of selected chromosomes equals to n .
-

Algorithm 2: Roulette-wheel selection Algorithm

3.3.4 Crossover

Biologically the crossing is when two chromosomes roll up and from there begins the process of mitosis, where the cell with the two chromosomes divides, obtaining information from both chromosomes. This inheritance mechanism is essential in genetic algorithms and allows defining how genetic information will pass from parents to children. The cross allows to fuse the genetic information of two individuals and pass it on to the children. Although there are different crossing methods in the literature, a two-point crossing was performed in our experiments. Figure 3.12 graphically shows the procedure for the crossover of two points.

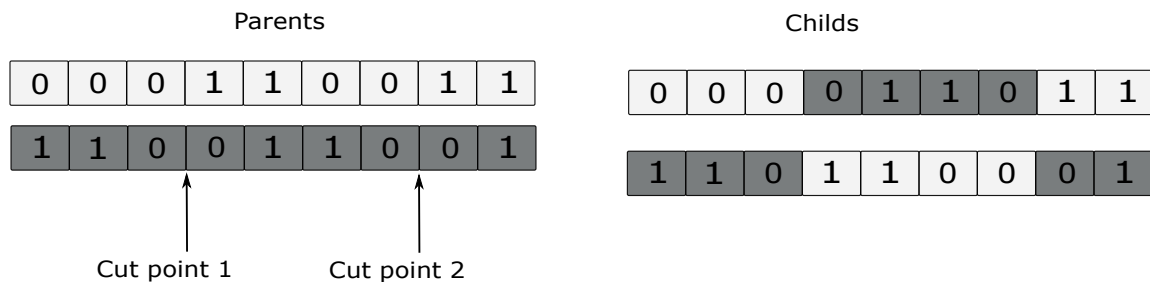


FIGURE 3.12: Two points crossover

3.3.5 Mutation

The mutation allows chromosomes to be modified randomly; this is replicated in the evolution when an error in reproduction, known as a copying error or a genes deformation, is generated. In genetic algorithms, mutation probability is usually much more significant than in any other biological organism. The mutation eventually allows a chromosome

to leap into the unknown by randomly modifying its chromosome. This operator is essential for the genetic algorithm as it prevents it from being trapped in local minima.

The mutation applies to a single gene; it is generally implemented with a low probability, such as 0.01 or 0.001. Mutation involves reversing the value of a bit on a chromosome. For example, with a mutation rate of 0.01, it might be expected that a gene on a 100-gene chromosome could be reversed.

3.3.6 Stop Condition

The stop condition is very important for a genetic algorithm. There are two regularly used stop conditions 1) when the genetic algorithm has reached the number of generations stipulated in advance and 2) when it is detected that most of the population has converged, causing its chromosomes to lack sufficient diversity. The first can cause the arrest to be carried out prematurely, and the solution obtained is not adequate, while the second only stops when the chromosomes are very similar to each other and to continue with the evolution is meaningless.

In the experimental results, the algorithm ends after ten iterations if there is no improvement in the evolution of the individuals. The string with the highest precision is stored with the obtained precision.

3.3.7 Elitism

Elitism is generally used in GA to avoid losing or eliminating the individual with the best fitness between the generations. This technique selects the fittest individual or fittest individuals and passes them on intact to the next generation. In each generation, the individual with the best aptitude is obtained and compared with the previous generation. If a chromosome is not obtained that improves the fitness obtained so far, the previous chain remains intact in the new generation.

In the proposed method, no pure elitism is used. We use a modified form in which a single chromosome (the one with the best fitness) from each generation is copied in the next generation.

Chapter 4

Experimental results

In this Chapter, we show the results obtained with the proposed algorithm and methodology. Each section describes the data sets and the parameters used to arrive at the results obtained. In addition, the comparisons obtained with other segmentation methods in state-of-the-art are shown.

4.1 Data sets

4.1.1 ICL Data set

The ICL leaf data set is a set of plant leaf images that have been widely used in recent years in many investigations [Cer+17] [Cer+18] [Wan+20] [ZWY17], is so-called because members of the Intelligent Computing Laboratory (ICL) collected it at the Institute of Intelligent Machines of the Chinese Academy of Sciences. It is probably the most extensive data set available for research.

The dataset contains 16851 images of 220 species. Each species has between 26 and 1078 examples. In all the images, the collectors took care that the background is white, which allows contrasting the colors of the sheet and helps to improve the quality of the segmentation. In addition, they also ensured that the lighting was adequate; this allows to reduce or eliminate parts with too much lighting, improving the quality of the segmentation. Fig. 4.1 shows different images of randomly selected plants.



FIGURE 4.1: ICL Dataset

4.1.2 Coffee diseases data set

The Coffee diseases data set was obtained from the data.mendeley.com website and collected by the authors of [Kro19]. It contains 739 images of healthy leaves and four common diseases of the coffee arabica species taken in coffee crops in Santa María de Marechal, Floreano, Brazil. All photos were taken in a controlled environment with a white background. The original images in the dataset have a resolution of 2048 pixels wide by 1024 pixels high.

Although all the images have a white background, the collectors did not take care that the lighting is adequate; this causes problems when segmenting. Fig. 4.1 shows different randomly selected images.



FIGURE 4.2: Coffee diseases data set

4.1.3 PlantVillage data set

The PlantVillage dataset was initially used in [HS15] consisting of 54303 images of healthy and unhealthy leaves divided into 38 categories by species and disease.

The PlantVillage dataset contains 38 classes and 54,305 images of 14 different plant species, 12 of which are healthy, 26 of which are diseased. The images in the dataset are color images of different sizes. In this data set, the background in the images varies very little. However, the collectors did not take care of the lighting, which increases the difficulty of adequately segmenting the images. Fig. 4.3 shows different plants and diseases selected at random.



FIGURE 4.3: PlantVillage dataset

4.1.4 LSM data set

The Mexican Sign Language (LSM) data set is a data set obtained by the authors to automatically identify and translate the LSM [EC+21]. The dataset contains videos and images of 249 words of the LSM. The authors use specific backgrounds and clothing to improve the quality of hand and face segmentation. It allows the authors to improve translation performance.

The authors collected 2,480 videos of sign words. The signals made in these videos came from eleven different people on average for each of the 249 words collected. Each video contains a single sign of the Mexican language. All the individuals are in front of the camera, with the left and right hands at rest. Then the individual starts the movement of the hand gesture, and finally, after that, the individual ends with hands at rest.

From the videos, the number of images obtained is 37,200 and on average 15 images represent one word. That is, the set of 15 images with hands and face positions describe a word from the LSM.



FIGURE 4.4: LSM dataset

4.2 Segmentation of data sets

This section shows the segmentation results obtained with the classical techniques in the different data sets used.

4.2.1 ICL data set

The ICL dataset is one of the best-developed sets of plant images. Most of the images have a white background, and a few images have a sky blue background. However, in the experimental results, this does not affect the quality of the segmentation. Figure 4.5 shows the results obtained with Otsu and PCA. The two methods proposed in this thesis were excluded since the images do not need elaborate algorithms to segment images. The use of basic segmentation algorithms such as Otsu and PCA allows images to be segmented appropriately with high quality. In Figure 4.5 the first row shows some randomly selected images, the second row shows the segmentation obtained with the Otsu segmentation technique, and the third row shows the results obtained using the PCA method when segmenting.

As can be seen, in this case, because the image collectors were concerned with having a white or blue background and adequate lighting, the use of basic algorithms allows obtaining a good quality of segmentation and quickly obtaining the region we need.

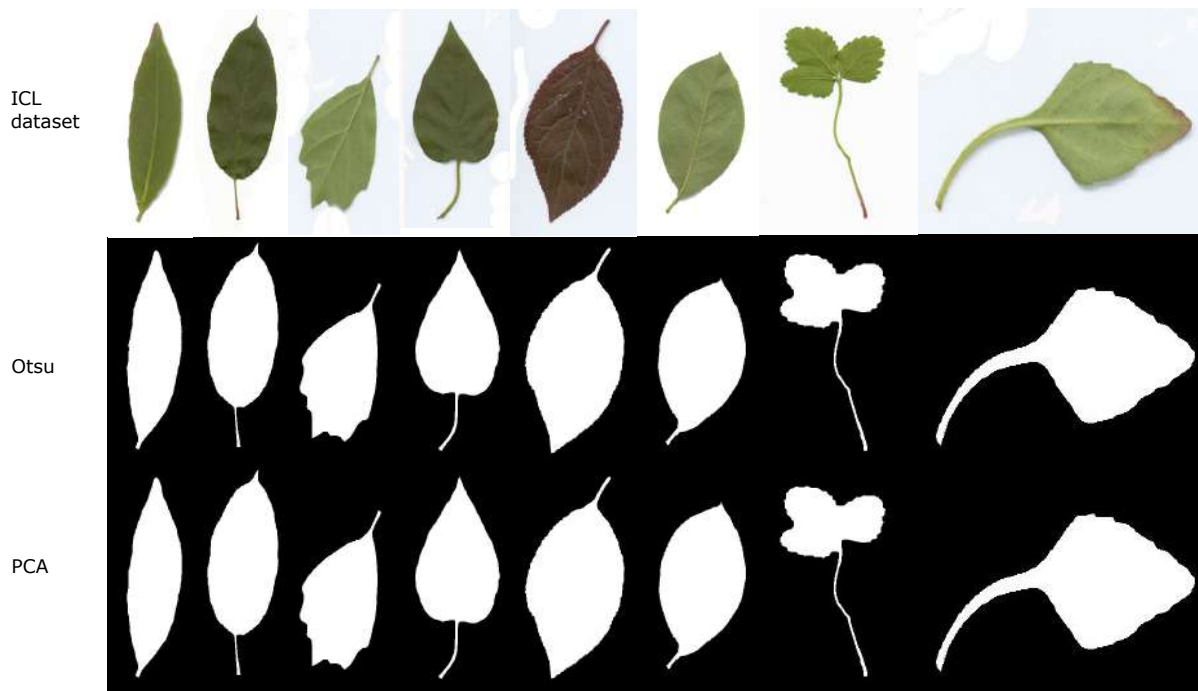


FIGURE 4.5: Segmentation of ICL data set

4.2.2 Coffee diseases data set

The second data set used is the coffee disease data set [Kro19]. Although all the images in the dataset also have a white background, the authors were not careful with the lighting, and the images have some shadows. These shadows present some problems for some basic segmentation techniques. Figure 4.6 shows four random images from the data set in the first row. The remaining rows show the segmentation results obtained with the different techniques used.

The second row shows the results obtained with the Otsu method. The Otsu method is one of the most widely used segmentation methods due to its performance. However, it has trouble segmenting some images with poor lighting, especially in the shadows of the image, or even confuses some areas within the sheet that are lighter than the rest of the sheet.

The third row shows the segmentation obtained by the PCA method. With the PCA method like Otsu. PCA still has trouble segmenting due to poor lighting.

The fourth and fifth columns show the segmentation results obtained with the proposed methods. In these cases, a better quality segmentation can be observed.

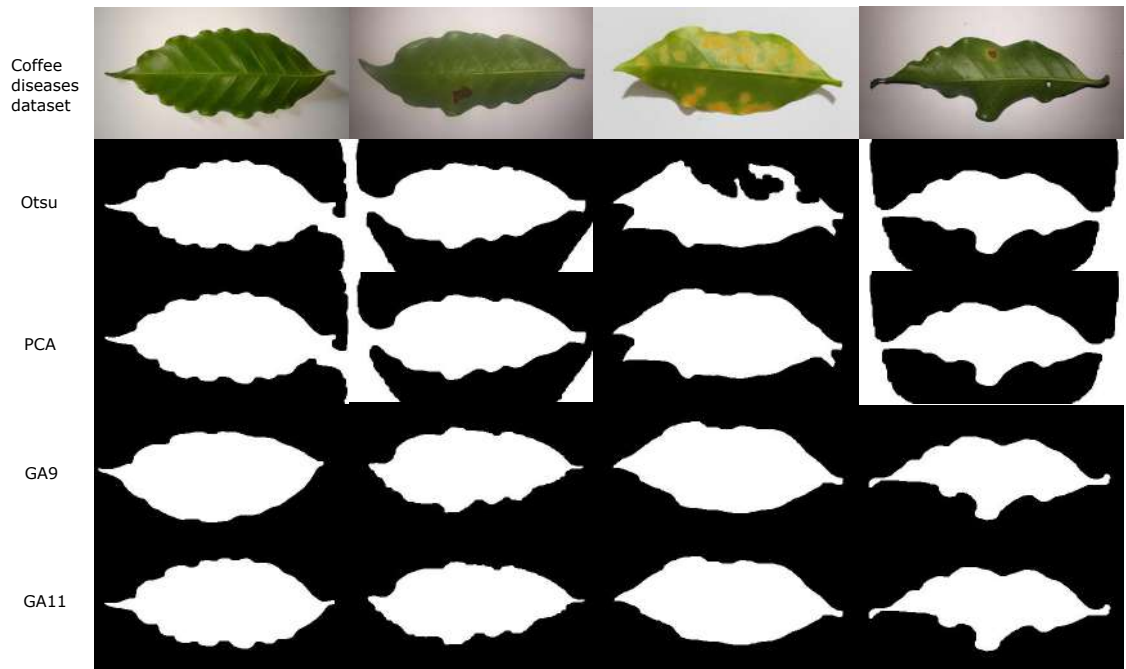


FIGURE 4.6: Segmentation of Coffee diseases data set

4.2.3 PlantVillage data set

This dataset has more trouble segmenting because the collectors were not careful in the lighting and, in some cases, the bottom of the sheet. Figure 4.7 shows the first row with the randomly selected images. Rows 2,3,4, and 5 show the images of row 1 segmented using Otsu, PCA, GA9, and GA11, respectively.

It is clear to see in the second and third rows that the Otsu and PCA algorithms do not perform well when there are lighting problems in the image. It affects the process too much because, in general, segmentation is a step in pattern recognition; obtaining a poorly segmented image will cause the extracted features to be naughty and, therefore, a poor recognition system. In addition, it is evident in the images that even some shadows are segmented as if they were part of the object to be segmented (Row 2, Column 5 and Row 2, Column 5). On the other hand, the proposed algorithms GA9 and GA11 show outstanding results when segmenting the images.

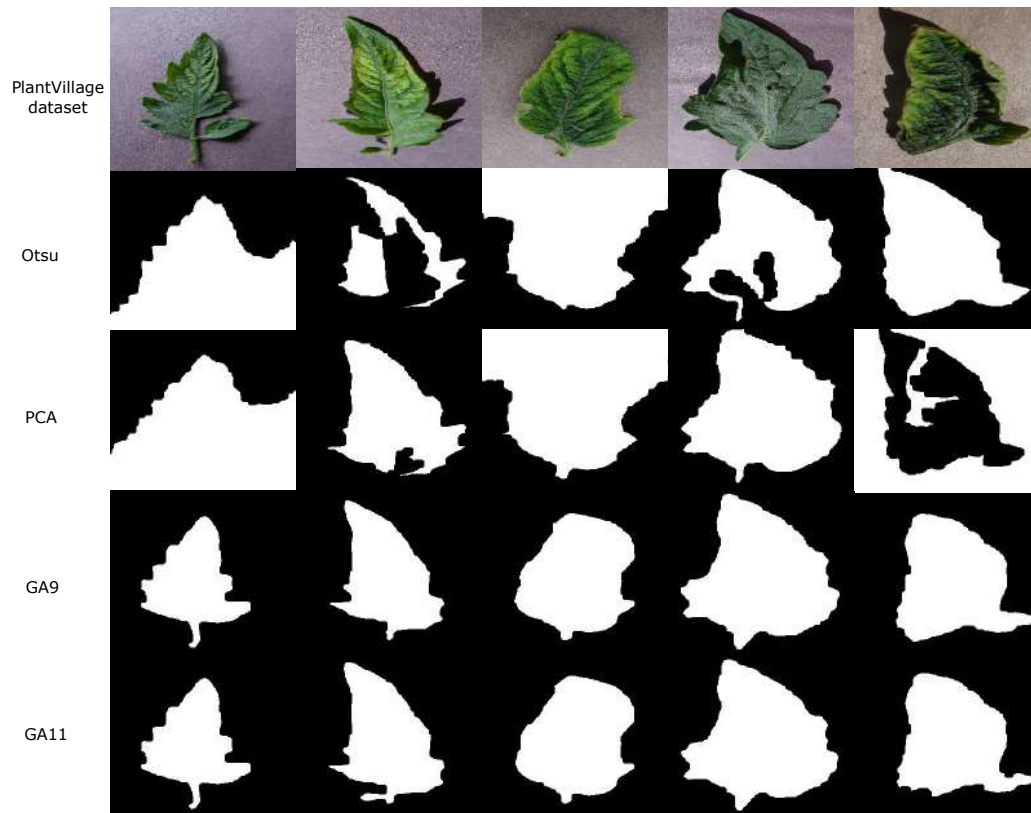


FIGURE 4.7: Segmentation of PlantVillage data set

4.2.4 LSM data set

Finally, the LSM data set contains images of the Mexican sign language. All images were taken with a black background and black clothing. The set of images is very easy to segment. However, some images have lighting problems—the first two images of row 1 in Figure 4.8—. The Otsu and PCA methods have problems segmenting the image, even the GA algorithm (it presents problems because it only segments the part of the face and not the hands). The good segmentation of hands is vital since the direction of one or both hands together with the position of one of the hands or contact of one of the hands with the body are patterns that are sought to identify the sign made.

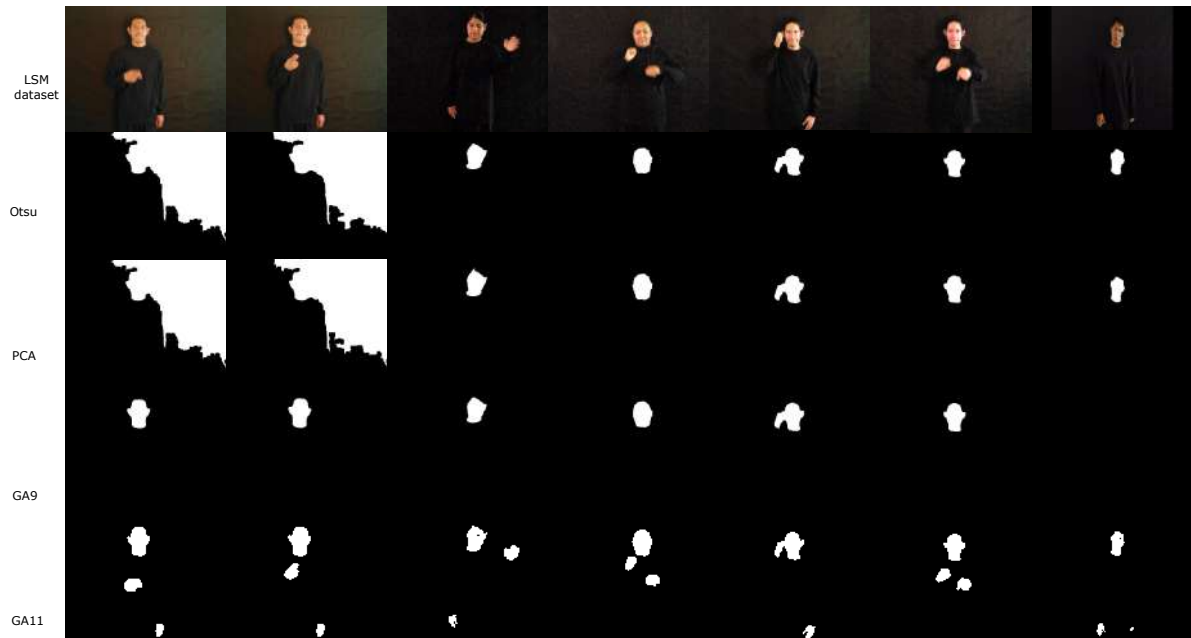


FIGURE 4.8: Segmentation of LSM dataset

Figure 4.8 shows that the segmentations obtained with Otsu and PCA fail to obtain the regions of the hands that are very important for the recognition of signs. Even the proposed algorithm GA9 has problems finding the set of suitable variables to segment the image. On the other hand, the AG11 algorithm obtains a better segmentation quality in all images.

4.3 Results and comparative analysis

In this Section, we show the results obtained using the genetic algorithm and the different comparison metrics.

4.3.1 Convergence of the genetic algorithm

In the experimental results, initial populations with 120 individuals were used. In most experiments, the genetic algorithm converged to a minimum before 80 generations. The graph 4.9 shows the average distance between individuals for each generation. It also shows how the distance between individuals is considerable, and as the generations increase, this distance between individuals is narrowed and reduced.

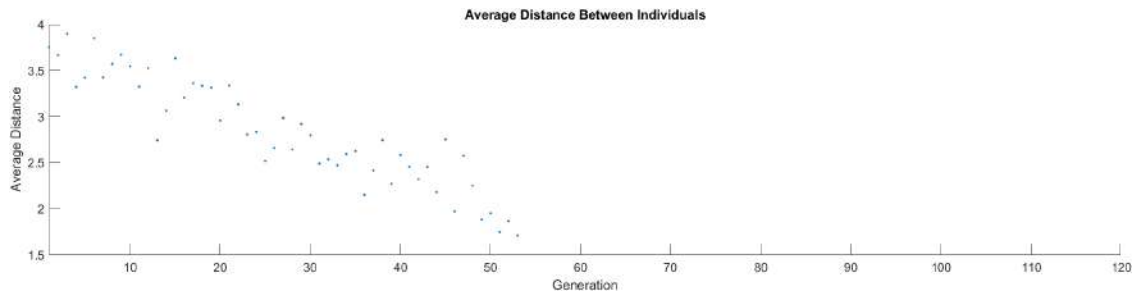


FIGURE 4.9: Average distance between individuals

Minimization was used in all experiments. Because Probabilistic Rand Index (PRI) is bounded between 0 and 1 with values of 1 as desired. Fitness is rescaled using $fitness = 10 - pri$, that is why the fitness of the best individual is 9,02846. In the graph, the points in blue represent the average fitness of the individuals. It should be mentioned that when an individual has very poor fitness, an extreme penalty was implemented in our algorithm to prevent the geneticist from selecting those individuals. The black dots on the graph represent the best individual in the generation. The Figure shows how the distances are shortened, and the penalty decreases in each generation.

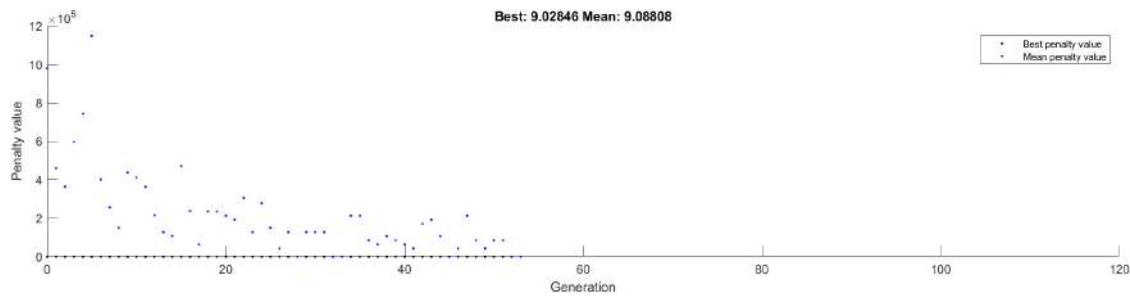


FIGURE 4.10: Best individual and average fitness of individuals

4.4 Discussions

This Section discusses the results obtained in the different data sets and using the Probabilistic Rand Index (PRI), Global Consistency Error (GCE), and The Variance of Information (VoI) metrics. In the thesis, 4 data sets were used. However, in this Section, only the results obtained in 3 data sets (Coffee diseases, PlantVillage, and LSM datasets) are discussed because the first data set (ICL) is very easy to segment, and the results were very similar to each other, they were excluded from this discussion.

4.4.1 Results obtained with the Coffee Diseases data set

For the Coffee diseases data set, the Otsu and PCA algorithms segment very well in some cases. However, when there are lighting problems, the poorly lit parts are segmented as part of the desired region. This pattern is repeated in many of the images. 16 random images were used for these comparisons. These were manually segmented and compared with the results obtained with the Otsu, PCA techniques, and the proposed GA9 and GA11 algorithms.

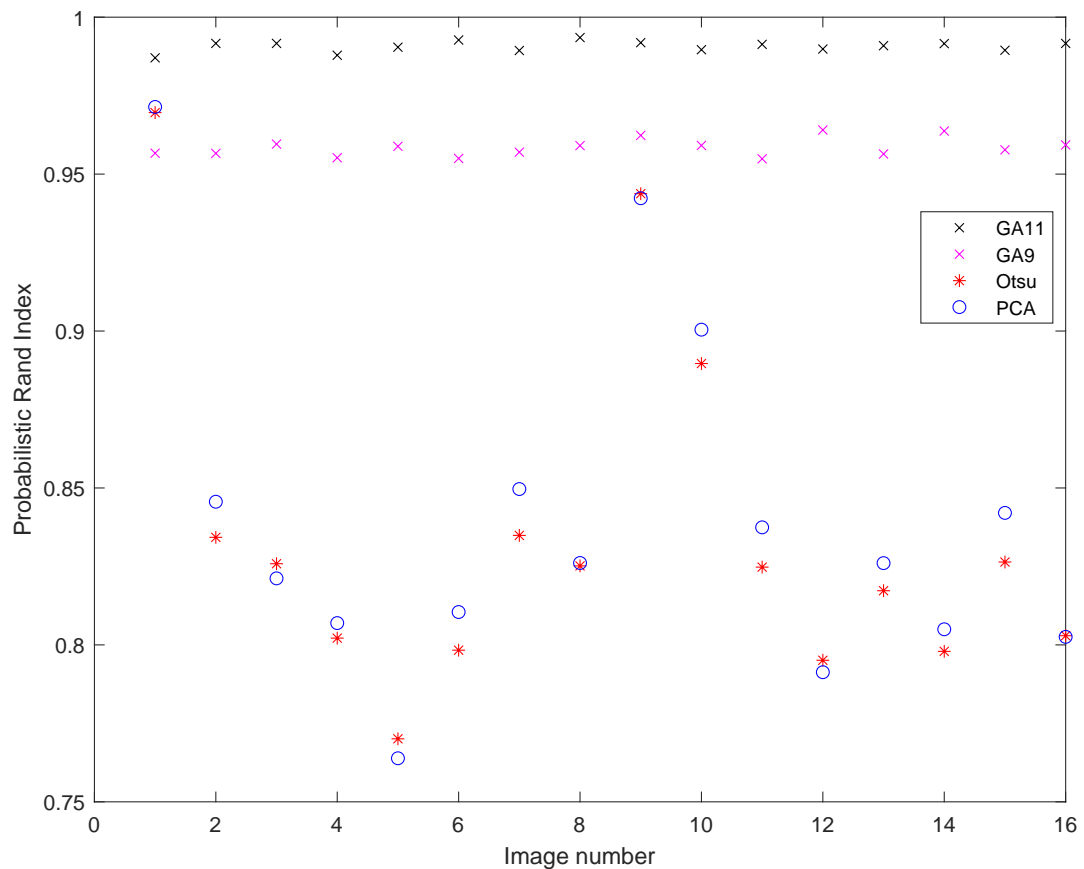


FIGURE 4.11: The Probabilistic Rand Index (PRI) for each image in Coffee diseases dataset

The PRI, GCE, and VoI metrics calculate the similarity between two segmented images. In the case of PRI, its range is between $[0, 1]$; closer to 1 represents that the segmentation obtained is very similar to the ground truth segmentation. Figure 4.11 shows the results obtained using the PRI comparison metric.

The Figure 4.11 show the PRI for each image with the different segmentation methods. The PRI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu

segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average PRI across all images was GA11: 0.9907, GA9: 0.9585, PCA: 0.8401, Otsu: 0.8349.

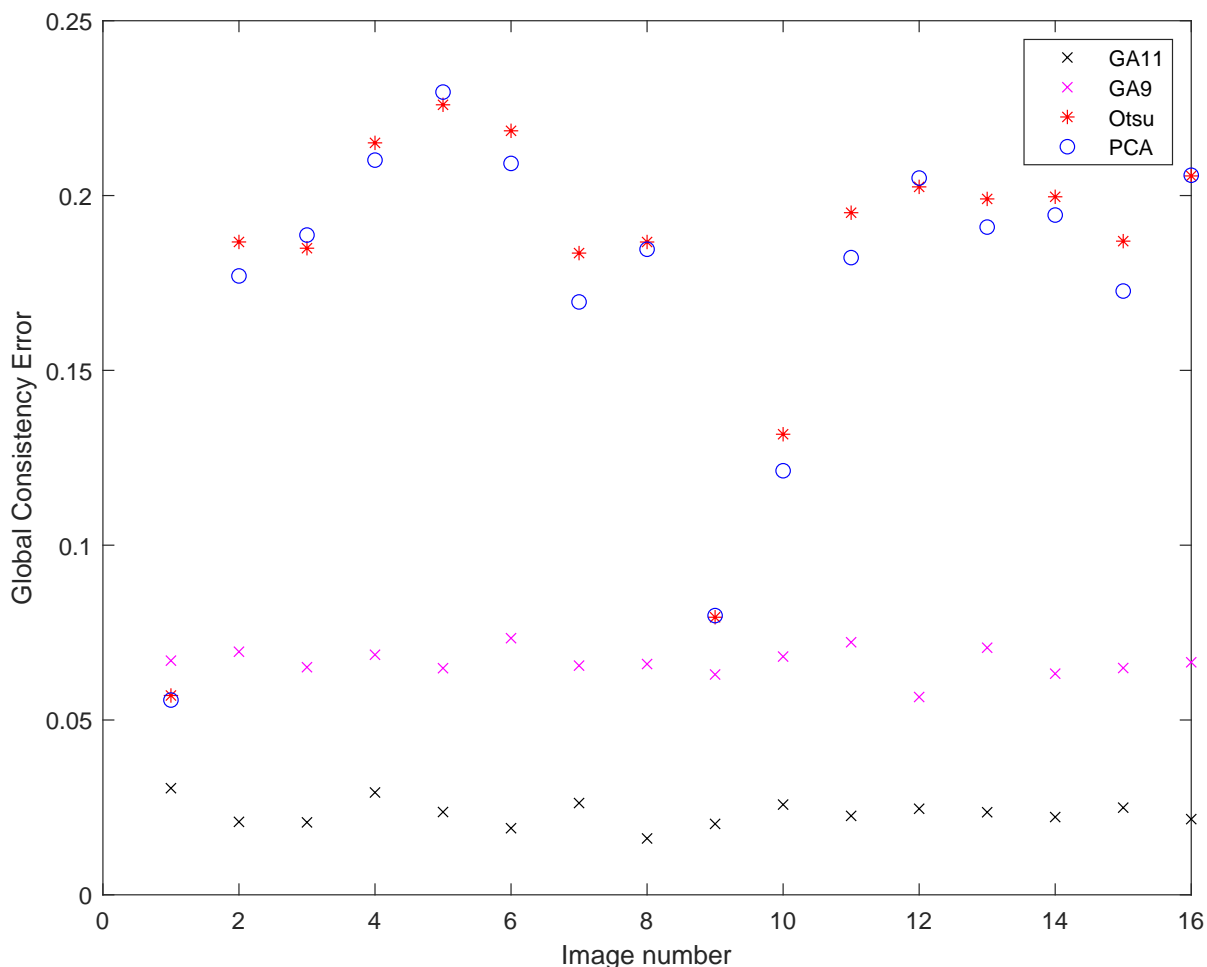


FIGURE 4.12: The Global Consistency Error (GCE) for each image in Coffee diseases dataset

In GCE cases, the range of values of the comparison metric is between $[0, 1]$ while closer to 0 indicates that the segmentation obtained is more similar to manual segmentation or Ground truth.

The Figure 4.12 show the GCE for each image with the different segmentation methods. The GCE for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average GCE across all images was GA11: 0.0233, GA9: 0.0667, PCA: 0.1736, Otsu: 0.1787.

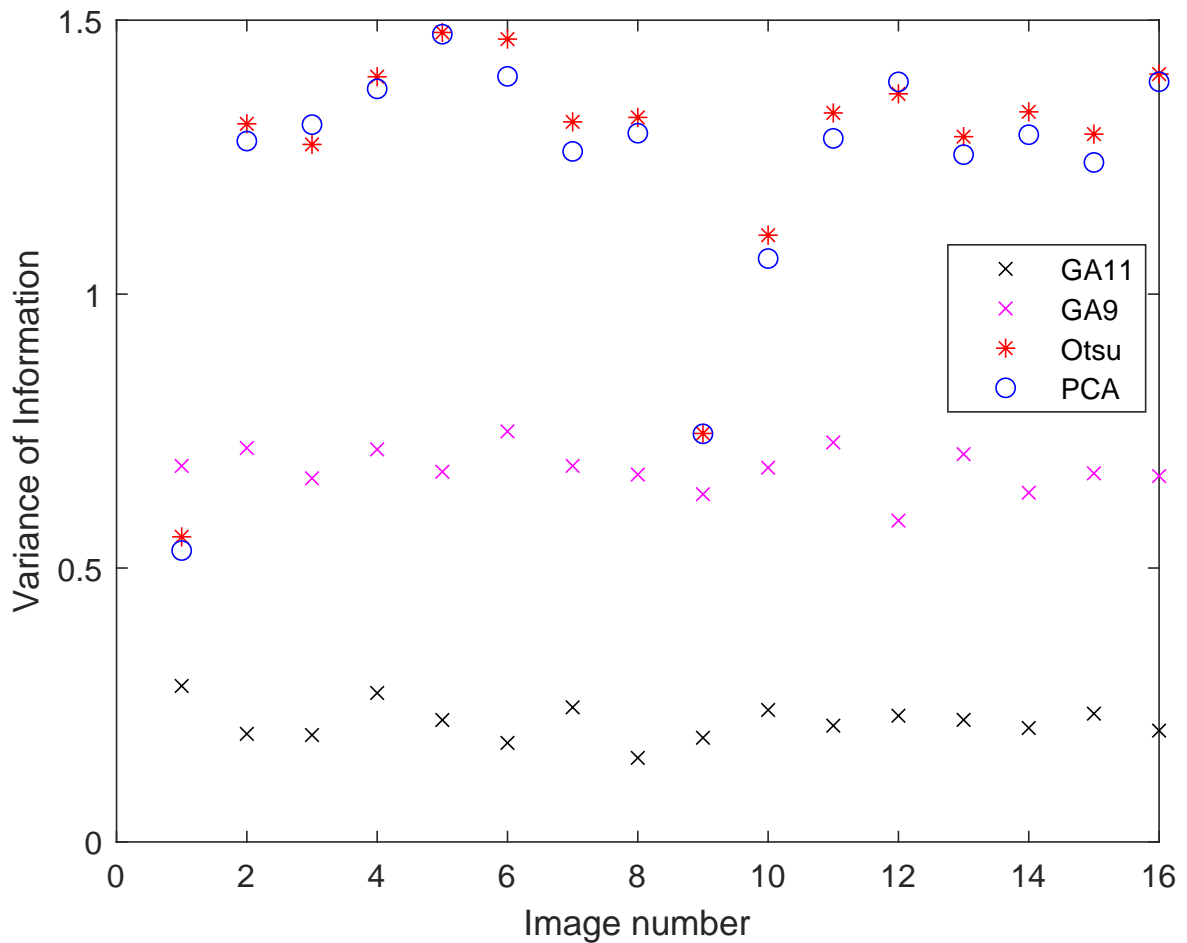


FIGURE 4.13: The Variance of Information (VoI) for each image in Coffee diseases dataset

In the case of Variance of Information, the range of values is $[0, +\infty]$, as in the case of GCE, the closer to 0 it indicates that the result of the segmentation obtained is similar to manual segmentation or Ground truth.

The Figure 4.13 show the VoI for each image with the different segmentation methods. The VoI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average VoI across all images was GA11: 0.2183, GA9: 0.6805, PCA: 1.2235, Otsu: 1.2488.

In this case, the Otsu and PCA algorithms segment the leaf well but add a part that does not belong to the leaf due to poor lighting. The segmentation algorithm must segment only the part of the image to extract better features and improve performance when classifying.

4.4.2 Results obtained with the PlantVillage dataset

For the PlantVillage dataset, segmentation is more difficult because the lighting was not fully taken care of, and segmentation techniques confuse the required region with shadows. Because many images have this problem, segmentation is generally very poor. One hundred eleven random images were used for these comparisons. These were manually segmented and compared with the results obtained with the Otsu, PCA techniques, and the proposed GA9 and GA11 algorithms.

The Figure 4.14 show the PRI metric for each image with the different segmentation methods. The PRI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average PRI across all images was GA11: 0.9566, GA9: 0.9475, PCA: 0.7446, Otsu: 0.6971.

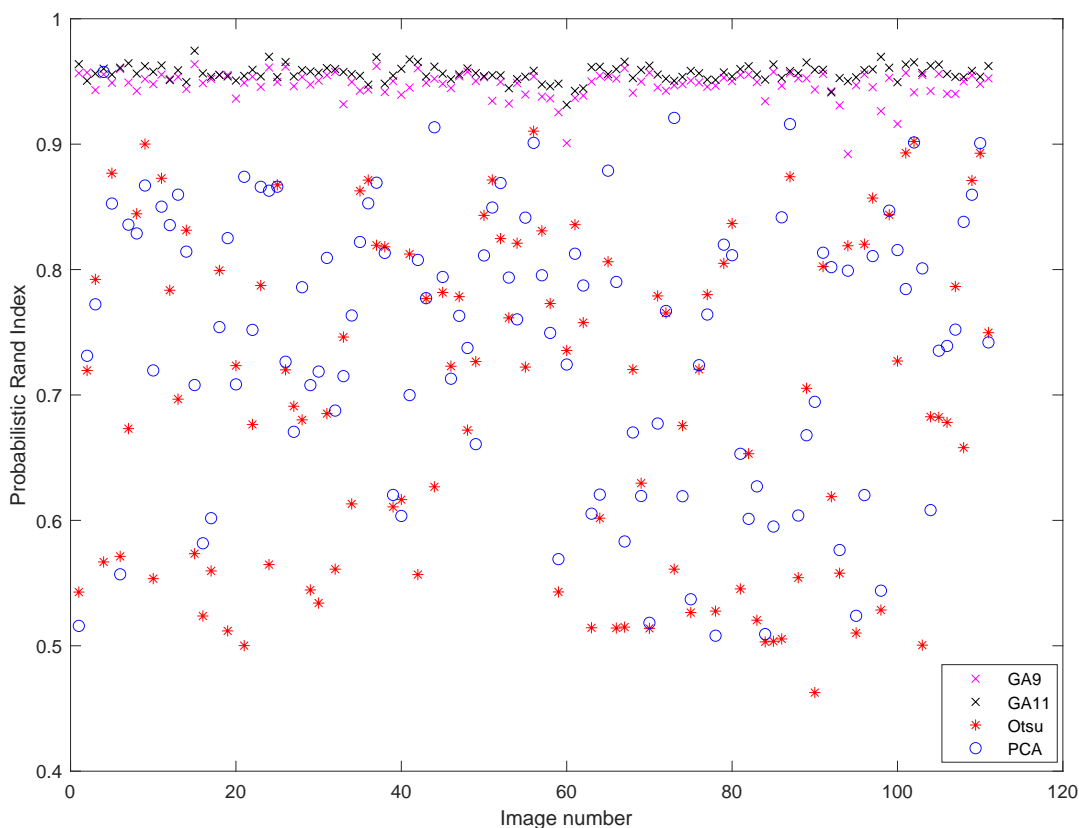


FIGURE 4.14: The Probabilistic Rand Index (PRI) for each image in PlantVillage dataset

The Figure 4.15 show the GCE for each image with the different segmentation methods.

The GCE for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average GCE across all images was GA11: 0.0742, GA9: 0.0808, PCA: 0.2466, Otsu: 0.2845.

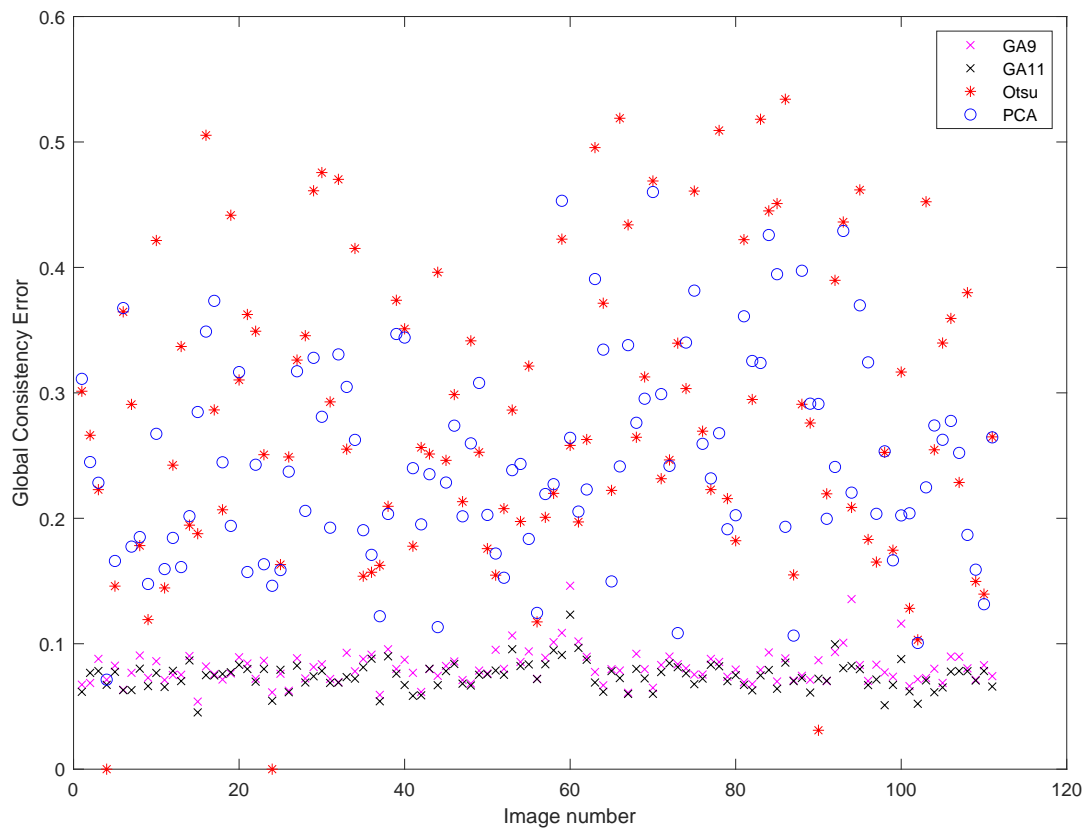


FIGURE 4.15: The Global Consistency Error (GCE) for each image in PlantVillage dataset

The Figure 4.16 show the VoI for each image with the different segmentation methods. The VoI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average VoI across all images was GA11: 0.7431, GA9: 0.8091, PCA: 1.7293, Otsu: 1.9585.

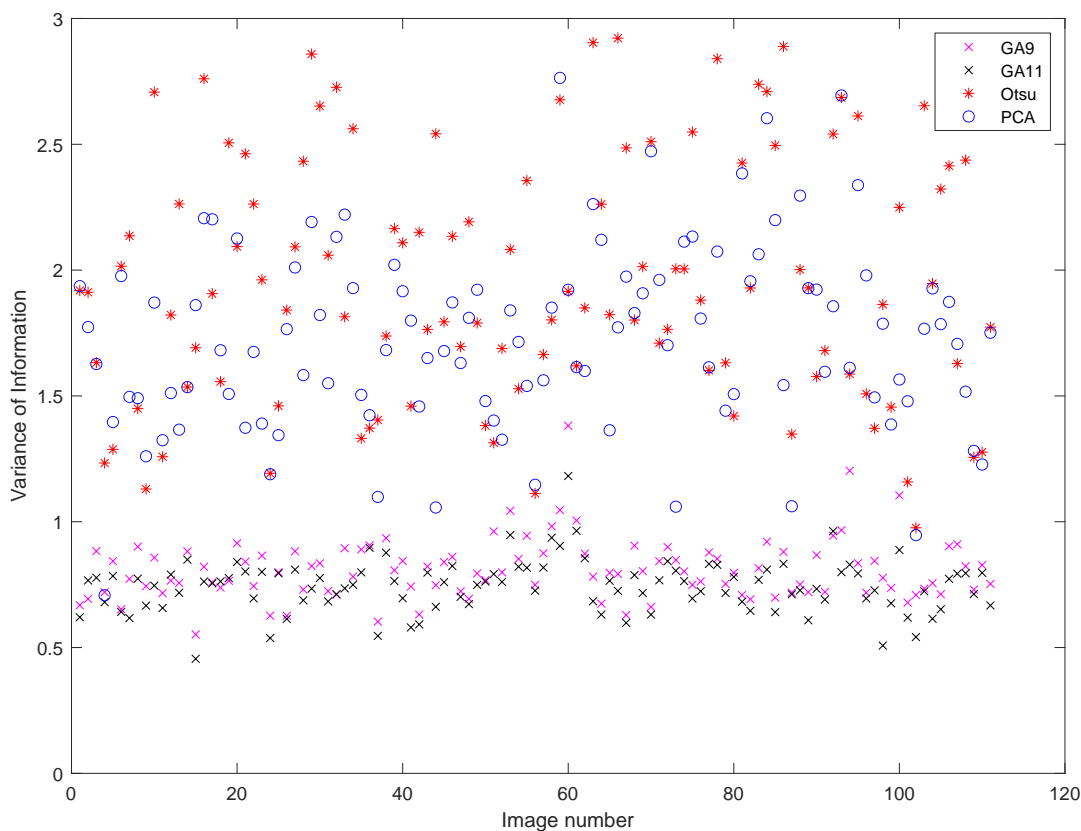


FIGURE 4.16: The Variance of Information (VoI) for each image in PlantVillage dataset

In the three graphs 4.14 4.15 4.16 it is easy to see how the results are better with the proposed techniques, this due to two factors: 1) the three metrics are always better and 2) the variance between the results is slight. It allows us to conclude that the segmentation results improve those obtained by the two classical techniques with which they were compared.

4.4.3 Results obtained with the LSM data set

For the LSM data sets, segmentation is even more difficult because there is not only one region to segment but several regions. It seems to be not well calculated or taken into account by the comparison metrics. In addition to the above, the data set used presents several lighting problems complicating the segmentation. Twenty-six random images were used for these comparisons. These were manually segmented and compared with the results obtained with the Otsu, PCA techniques, and the proposed GA9 and GA11 algorithms.

The Figure 4.14 show the PRI metric for each image with the different segmentation methods. The PRI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average PRI across all images was GA11: 0.9925, GA9: 0.9668, PCA: 0.9701, Otsu: 0.9378.

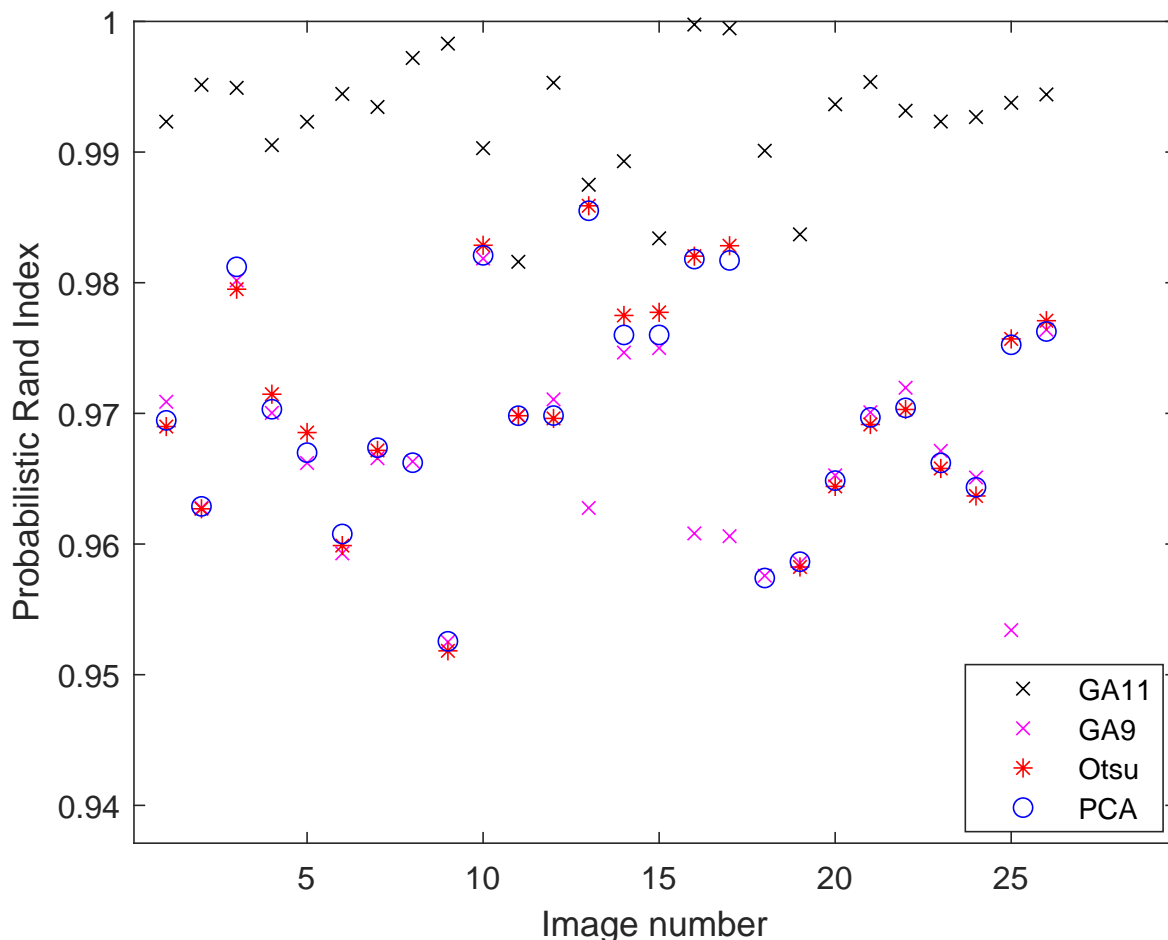


FIGURE 4.17: The Probabilistic Rand Index (PRI) for each image in LSM dataset

The Figure 4.14 show the PRI metric for each image with the different segmentation methods. The PRI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average PRI across all images was GA11: 0.0240, GA9: 0.0316, PCA: 0.0185, Otsu: 0.0394.

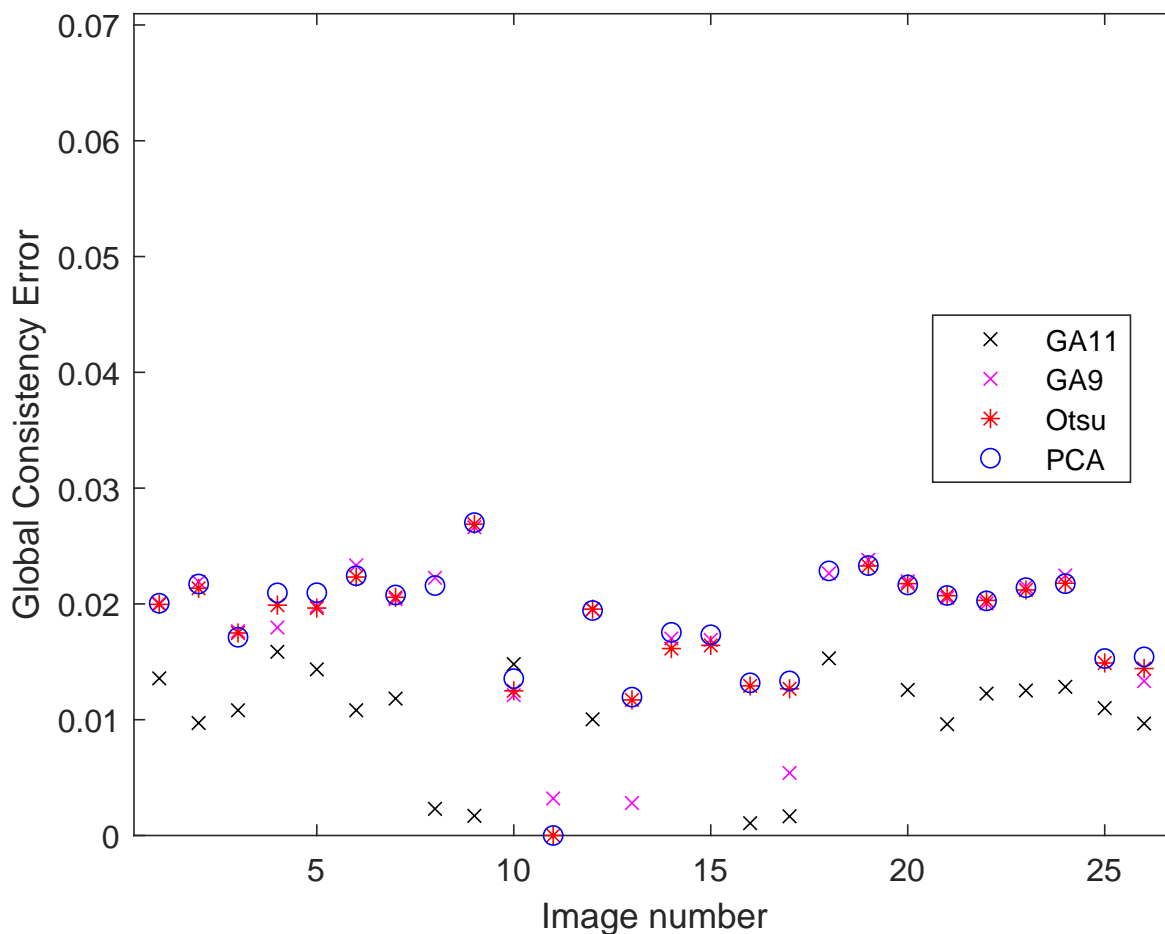


FIGURE 4.18: The Global Consistency Error (GCE) for each image in LSM dataset

The Figure 4.14 show the PRI metric for each image with the different segmentation methods. The PRI for the proposed methods GA9 (magenta 'x') and GA11 (black 'x') and the Otsu segmentation algorithm (red '*') and the segmentation based on PCA (blue circles) were measured and compared all the images with the ground truth dataset. The average PRI across all images was GA11: 0.1198, GA9: 0.2341, PCA: 0.2310, Otsu: 0.3150.

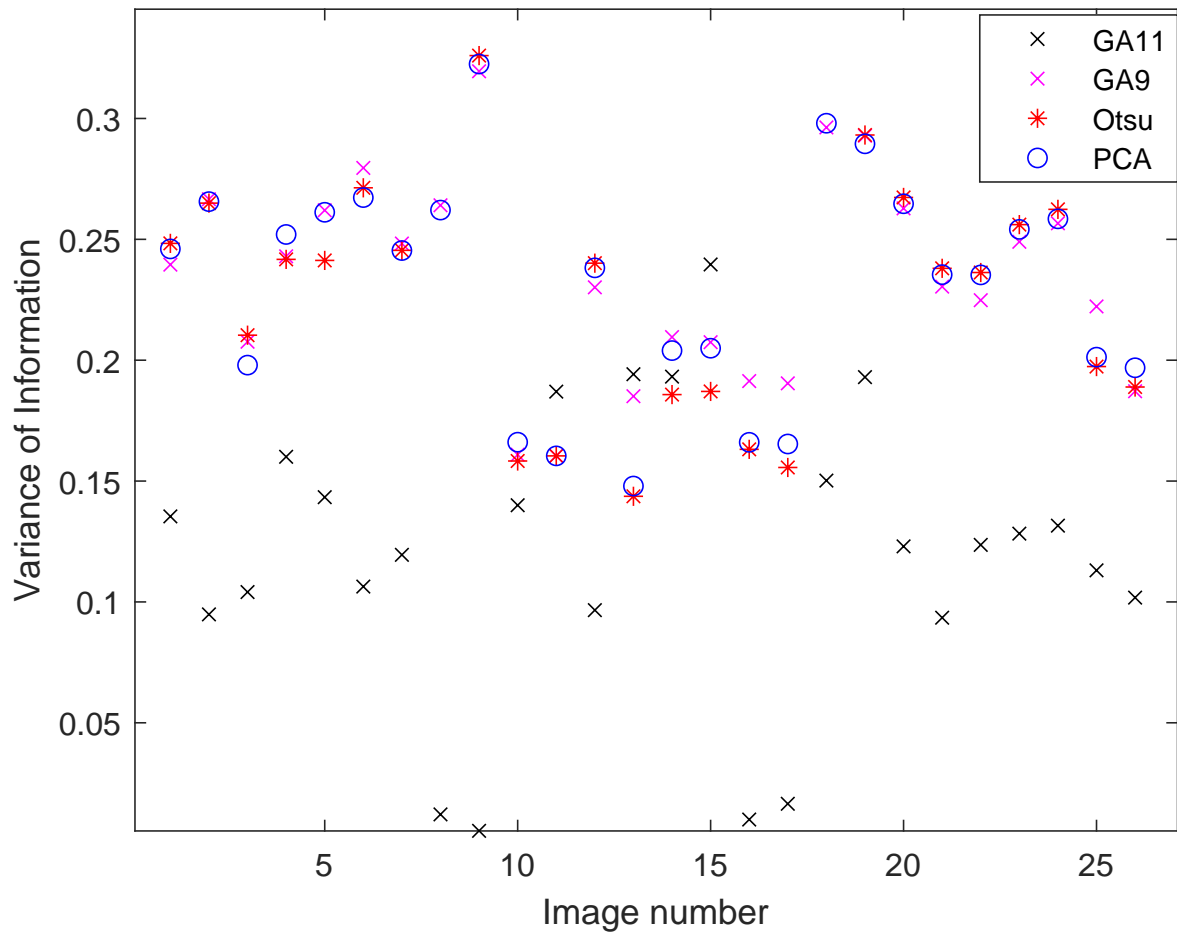


FIGURE 4.19: The Variance of Information (VoI) for each image in LSM dataset

In this data set, the results obtained with PCA are even better than those obtained with the proposed GA9 algorithm; several tests were performed to verify results, and the results are very similar. Those reported here are a sample of those obtained.

Chapter 5

Conclusions

In this Thesis, two new image segmentation algorithms for datasets were proposed. The proposed algorithms permit to get the best combination of variables and segmentation algorithms to segment an image. The proposed algorithms were proved and compared with Otsu, and segmentation based on PCA. Four sets of images were tested in the experiments carried out. The experiments carried out show how bad lighting affects the basic algorithms, while the proposed algorithms find a good combination of variables to segment.

The calculation of the identification of regions of an image that are homogeneous and connected is considered a combinatorial search problem that is very time consuming, and sometimes the definitions of these regions are inadequate. GAs are effective in the case of combinatorial problems as they allow parallel exploration of the search space. Consequently, this Thesis proposed a new approach to segmentation based on AG that can improve computational time and give us a better segmentation quality. The method consists of randomly testing a whole generation of possible solutions, selecting the best options, and then improving them through evolution; when the necessary parameters are found, they are applied to the remaining set of images. It was observed that errors could occur when classifying the background light as pixels of interest by individually applying various segmentation methods to an image. The combination of preprocessing and morphological operators reduces these problems.

With the analysis of the main segmentation techniques and their respective classification, we understand that there is no perfect technique for image segmentation. Each image is individual and has its characteristics. For this reason, a single method cannot be generalized. In most cases, the result of a good segmentation will combine several techniques, operators and preprocesses. With the above in mind, automatic segmentation genetic algorithms can give better results, reducing costs and speeding up the process.

Future work should consider using more and more variable segmentation algorithms

to improve segmentation quality in more complex image sets. It is also essential to study the comparison metrics of the segmented images and propose new performance metrics. Especially in sets of images with several required regions. Even adding penalty factors for small but significant regions.

Bibliography

- [Abb+11] M. Abbasgholipour **and others**. ?Color image segmentation with genetic algorithm in a raisin sorting system based on machine vision in variable conditions? *in Expert Systems with Applications*: 38.4 (2011), **pages** 3671–3678. DOI: [10.1016/j.eswa.2010.09.023](https://doi.org/10.1016/j.eswa.2010.09.023).
- [AF+13] Ali Qusay Al-Faris **and others**. ?Breast MRI Tumour Segmentation Using Modified Automatic Seeded Region Growing Based on Particle Swarm Optimization Image Clustering? *in Advances in Intelligent Systems and Computing*: Springer International Publishing, 2013, **pages** 49–60. DOI: [10.1007/978-3-319-00930-8_5](https://doi.org/10.1007/978-3-319-00930-8_5).
- [Aro+08] S. Arora **and others**. ?Multilevel thresholding for image segmentation through a fast statistical recursive algorithm? *in Pattern Recognition Letters*: 29.2 (2008), **pages** 119–125. DOI: [10.1016/j.patrec.2007.09.005](https://doi.org/10.1016/j.patrec.2007.09.005).
- [AS13] AidinSakhavati Akbar ShahrzadKhashandarag MirkamalMirnia. ?A New Method for Medical Image Clustering Using Genetic Algorithm? *in IJCSI International Journal of Computer Science Issues*: 10.1 (**january** 2013).
- [BLM95] Bir Bhanu, Sungkee Lee **and** John Ming. ?Adaptive image segmentation using a genetic algorithm? *in IEEE Transactions on Systems, Man, and Cybernetics*: 25.12 (1995), **pages** 1543–1567. DOI: [10.1109/21.478444](https://doi.org/10.1109/21.478444).
- [BM03] Vitoantonio Bevilacqua **and** Giuseppe Mastronardi. ?Image Segmentation Using a Genetic Algorithm? *in Soft Computing Applications*: Physica-Verlag HD, 2003, **pages** 115–126. DOI: [10.1007/978-3-7908-1768-3_11](https://doi.org/10.1007/978-3-7908-1768-3_11).
- [BY11] O. Banimelhem **and** Ya Yahya. ?Multi-Thresholding Image Segmentation Using Genetic Algorithm? *in* 2011.
- [Can86] John Canny. ?A Computational Approach to Edge Detection? *in IEEE Transactions on Pattern Analysis and Machine Intelligence*: PAMI-8.6 (1986), **pages** 679–698. DOI: [10.1109/tpami.1986.4767851](https://doi.org/10.1109/tpami.1986.4767851).

- [Cer+17] Jair Cervantes **and others**. ?Análisis Comparativo de las técnicas utilizadas en un Sistema de Reconocimiento de Hojas de Planta? **in** *Revista Iberoamericana de Automática e Informática Industrial RIAI*: 14.1 (2017), **pages** 104–114. DOI: [10.1016/j.riai.2016.09.005](https://doi.org/10.1016/j.riai.2016.09.005).
- [Cer+18] Jair Cervantes **and others**. ?Complex Identification of Plants from Leaves? **in** *Intelligent Computing Methodologies*: Springer International Publishing, 2018, **pages** 376–387. DOI: [10.1007/978-3-319-95957-3_41](https://doi.org/10.1007/978-3-319-95957-3_41).
- [CM+20] Miguel Á. Castillo-Martínez **and others**. ?Color index based thresholding method for background and foreground segmentation of plant images? **in** *Computers and Electronics in Agriculture*: 178 (2020), **page** 105783. DOI: [10.1016/j.compag.2020.105783](https://doi.org/10.1016/j.compag.2020.105783).
- [CM20] Shouvik Chakraborty **and** Kalyani Mali. ?Fuzzy Electromagnetism Optimization (FEMO) and its application in biomedical image segmentation? **in** *Applied Soft Computing*: 97 (2020), **page** 106800. DOI: [10.1016/j.asoc.2020.106800](https://doi.org/10.1016/j.asoc.2020.106800).
- [CS10] Toby Breckon Chris Solomon. *Fundamentals of Digital Image Processing*. John Wiley Sons Inc, 20 **december** 2010. 344 **pagetotals**. ISBN: 0470844736. URL: https://www.ebook.de/de/product/3996688/chris_solomon_toby_breckon_fundamentals_of_digital_image_processing.html.
- [DBB] I. Diaz, J. Branch **and** P. Boulanger. ?A Genetic Algorithm to Segment Range Image by Edge Detection? **in** *2005 International Conference on Industrial Electronics and Control Applications*: IEEE. DOI: [10.1109/icieca.2005.1644380](https://doi.org/10.1109/icieca.2005.1644380).
- [DBD16] Sourav De, Siddhartha Bhattacharyya **and** Paramartha Dutta. ?Automatic magnetic resonance image segmentation by fuzzy intercluster hostility index based genetic algorithm: An application? **in** *Applied Soft Computing*: 47 (2016), **pages** 669–683. DOI: [10.1016/j.asoc.2016.05.042](https://doi.org/10.1016/j.asoc.2016.05.042).
- [DD20] Naveen Singh Dagar **and** Pawan Kumar Dahiya. ?Edge Detection Technique using Binary Particle Swarm Optimization? **in** *Procedia Computer Science*: 167 (2020), **pages** 1421–1436. DOI: [10.1016/j.procs.2020.03.353](https://doi.org/10.1016/j.procs.2020.03.353).
- [DO08] Z DOKUR **and** T OLMEZ. ?Tissue segmentation in ultrasound images by using genetic algorithms? **in** *Expert Systems with Applications*: 34.4 (2008), **pages** 2739–2746. DOI: [10.1016/j.eswa.2007.05.002](https://doi.org/10.1016/j.eswa.2007.05.002).

- [DTV18] Antonios Danelakis, Theoharis Theoharis **and** Dimitrios A. Verganelakis. ?Survey of automated multiple sclerosis lesion segmentation techniques on magnetic resonance imaging? **in***Computerized Medical Imaging and Graphics*: 70 (2018), **pages** 83–100. DOI: [10.1016/j.compmedimag.2018.10.002](https://doi.org/10.1016/j.compmedimag.2018.10.002).
- [Dua+06] Abraham Duarte **and** others. ?Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic? **in***Pattern Recognition Letters*: 27.11 (2006), **pages** 1239–1251. DOI: [10.1016/j.patrec.2005.07.022](https://doi.org/10.1016/j.patrec.2005.07.022).
- [EC+21] Josué Espejel-Cabrera **and** others. ?Mexican sign language segmentation using color based neuronal networks to detect the individual skin color? **in***Expert Systems with Applications*: 183 (2021), **page** 115295. DOI: [10.1016/j.eswa.2021.115295](https://doi.org/10.1016/j.eswa.2021.115295).
- [Fre10] Daniel Freedman. ?An improved image graph for semi-automatic segmentation? **in***Signal, Image and Video Processing*: 6.4 (2010), **pages** 533–545. DOI: [10.1007/s11760-010-0181-9](https://doi.org/10.1007/s11760-010-0181-9).
- [GFHL20] Wilfrido Gómez-Flores **and** Juanita Hernández-López. ?Automatic adjustment of the pulse-coupled neural network hyperparameters based on differential evolution and cluster validity index for image segmentation? **in***Applied Soft Computing*: 97 (2020), **page** 105547. DOI: [10.1016/j.asoc.2019.105547](https://doi.org/10.1016/j.asoc.2019.105547).
- [GH88] David E. Goldberg **and** John H. Holland. **in***Machine Learning*: 3.2/3 (1988), **pages** 95–99. DOI: [10.1023/a:1022602019183](https://doi.org/10.1023/a:1022602019183).
- [GL+18] Farid Garcia-Lamont **and** others. ?Segmentation of images by color features: A survey? **in***Neurocomputing*: 292 (2018), **pages** 1–27. DOI: [10.1016/j.neucom.2018.01.091](https://doi.org/10.1016/j.neucom.2018.01.091).
- [Gon02] Rafael Gonzalez. *Digital image processing*. Upper Saddle River, N.J: Prentice Hall, 2002. ISBN: 0201180758.
- [GSJ20] Jatin Gupta, Sumindar Kaur Saini **and** Mamta Juneja. ?Survey of denoising and segmentation techniques for MRI images of prostate for improving diagnostic tools in medical applications? **in***Materials Today: Proceedings*: 28 (2020), **pages** 1667–1672. DOI: [10.1016/j.matpr.2020.05.023](https://doi.org/10.1016/j.matpr.2020.05.023).

- [HS15] David. P. Hughes **and** Marcel Salathe. ?An open access repository of images on plant health to enable the development of mobile disease diagnostics? **in**(25 november 2015): arXiv: [1511.08060](https://arxiv.org/abs/1511.08060) [[cs.CY](#)].
- [Hu+12] Rongxiang Hu **and**others. ?Multiscale Distance Matrix for Fast Plant Leaf Recognition? **in***IEEE Transactions on Image Processing*: 21.11 (2012), **pages** 4667–4672. DOI: [10.1109/tip.2012.2207391](https://doi.org/10.1109/tip.2012.2207391).
- [Huy+20] Bram Huygens **and**others. ?Application of evolutionary algorithms to optimise one- and two-dimensional gradient chromatographic separations? **in***Journal of Chromatography A*: 1628 (2020), **page** 461435. DOI: [10.1016/j.chroma.2020.461435](https://doi.org/10.1016/j.chroma.2020.461435).
- [Ito+06] Seiji Ito **and**others. ?An image segmentation method using histograms and the human characteristics of HSI color space for a scene image? **in***Artificial Life and Robotics*: 10.1 (2006), **pages** 6–10. DOI: [10.1007/s10015-005-0352-x](https://doi.org/10.1007/s10015-005-0352-x).
- [Jan+13] K. Janc **and**others. ?Genetic algorithms as a useful tool for trabecular and cortical bone segmentation? **in***Computer Methods and Programs in Biomedicine*: 111.1 (2013), **pages** 72–83. DOI: [10.1016/j.cmpb.2013.03.012](https://doi.org/10.1016/j.cmpb.2013.03.012).
- [JV20] S. Jose **and** C. Vijayalakshmi. ?Design and Analysis of Multi-Objective Optimization Problem Using Evolutionary Algorithms.? **in***Procedia Computer Science*: 172 (2020), **pages** 896–899. DOI: [10.1016/j.procs.2020.05.129](https://doi.org/10.1016/j.procs.2020.05.129).
- [Kha14] Muhammad Waseem Khan. ?A Survey: Image Segmentation Techniques? **in***International Journal of Future Computer and Communication*: (2014), **pages** 89–93. DOI: [10.7763/ijfcc.2014.v3.274](https://doi.org/10.7763/ijfcc.2014.v3.274).
- [KR20] Akshi Kumar **and** Sahil Raheja. ?Edge Detection using Guided Image Filtering and Enhanced Ant Colony Optimization? **in***Procedia Computer Science*: 173 (2020), **pages** 8–17. DOI: [10.1016/j.procs.2020.06.003](https://doi.org/10.1016/j.procs.2020.06.003).
- [Kro19] Renato A. Krohling. *BRACOL - A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests*. 2019. DOI: [10.17632/YY2K5Y8MXG.1](https://doi.org/10.17632/YY2K5Y8MXG.1).

- [KS19] Paulo Roberto Gardel Kurka **and** Aldo André Díaz Salazar. ?Applications of image processing in robotics and instrumentation? *in* *Mechanical Systems and Signal Processing*: 124 (2019), **pages** 142–169. DOI: [10.1016/j.ymsp.2019.01.015](https://doi.org/10.1016/j.ymsp.2019.01.015).
- [Law+82] K. Laws **and** others. ?The Phoenix Image Segmentation System: Description and Evaluation? *in* 1982.
- [Man+14] S. Manikandan **and** others. ?Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm? *in* *Measurement*: 47 (2014), **pages** 558–568. DOI: [10.1016/j.measurement.2013.09.031](https://doi.org/10.1016/j.measurement.2013.09.031).
- [Mar+] D. Martin **and** others. ?A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics? *in* *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*: IEEE Comput. Soc. DOI: [10.1109/iccv.2001.937655](https://doi.org/10.1109/iccv.2001.937655).
- [Mar03] David Royal Martin. ?An Empirical Approach to Grouping and Segmentation? phdthesis. EECS Department, University of California, Berkeley, 2003. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2003/5252.html>.
- [Mei05] Marina Meilă. ?Comparing clusterings? *in* *Proceedings of the 22nd international conference on Machine learning - ICML '05*: ACM Press, 2005. DOI: [10.1145/1102351.1102424](https://doi.org/10.1145/1102351.1102424).
- [MEKS20] Seyed Jalaleddin Mousavirad, Hossein Ebrahimpour-Komleh **and** Gerald Schaefer. ?Automatic clustering using a local search-based human mental search algorithm for image segmentation? *in* *Applied Soft Computing*: 96 (2020), **page** 106604. DOI: [10.1016/j.asoc.2020.106604](https://doi.org/10.1016/j.asoc.2020.106604).
- [MHS16] Sharada P. Mohanty, David P. Hughes **and** Marcel Salathé. ?Using Deep Learning for Image-Based Plant Disease Detection? *in* *Frontiers in Plant Science*: 7 (2016). DOI: [10.3389/fpls.2016.01419](https://doi.org/10.3389/fpls.2016.01419).
- [Nin+10] Jifeng Ning **and** others. ?Interactive image segmentation by maximal similarity based region merging? *in* *Pattern Recognition*: 43.2 (2010), **pages** 445–456. DOI: [10.1016/j.patcog.2009.03.004](https://doi.org/10.1016/j.patcog.2009.03.004).

- [Ots79] Nobuyuki Otsu. ?A Threshold Selection Method from Gray-Level Histograms? **in***IEEE Transactions on Systems, Man, and Cybernetics*: 9.1 (1979), **pages** 62–66. DOI: [10.1109/tsmc.1979.4310076](https://doi.org/10.1109/tsmc.1979.4310076).
- [OY18] Pedro Ventura de Oliveira **and** Keiji Yamanaka. ?Image Segmentation Using Multilevel Thresholding and Genetic Algorithm: An Approach? **in***2018 2nd International Conference on Data Science and Business Analytics (ICDSBA)*: IEEE, 2018. DOI: [10.1109/icdsba.2018.00078](https://doi.org/10.1109/icdsba.2018.00078).
- [Pap+05] Stelios E. Papadakis **and** others. ?A Genetic Based Approach to the Type I Structure Identification Problem? **in***Informatica*: 16.3 (2005), **pages** 365–382. DOI: [10.15388/informatica.2005.104](https://doi.org/10.15388/informatica.2005.104).
- [PB15] B.D. Phulpagar **and** R.S. Bichkar. ?Segmentation of Images Containing Multiple Intensity Levels using Genetic Algorithms? **in***International Journal of Computer Applications*: 126.9 (2015), **pages** 29–39. DOI: [10.5120/ijca2015906186](https://doi.org/10.5120/ijca2015906186).
- [Pun11] Thakare Punam; ?A Study of Image Segmentation and EdgeDetection Techniques? **in***International Journal on Computer Science and Engi-neering (IJCSE)*: vol 3 (2011), **pages** 899–905.
- [PZL09] Zhenkui Pei, Yanli Zhao **and** Zhen Liu. ?Image segmentation based on Differential Evolution algorithm? **in***2009 International Conference on Image Analysis and Signal Processing*: IEEE, 2009. DOI: [10.1109/iasp.2009.5054643](https://doi.org/10.1109/iasp.2009.5054643).
- [PZY10] Bo Peng, Lei Zhang **and** Jian Yang. ?Iterated Graph Cuts for Image Segmentation? **in***Computer Vision – ACCV 2009*: Springer Berlin Heidelberg, 2010, **pages** 677–686. DOI: [10.1007/978-3-642-12304-7_64](https://doi.org/10.1007/978-3-642-12304-7_64).
- [Ran71] William M. Rand. ?Objective Criteria for the Evaluation of Clustering Methods? **in***Journal of the American Statistical Association*: 66.336 (1971), **pages** 846–850. DOI: [10.1080/01621459.1971.10482356](https://doi.org/10.1080/01621459.1971.10482356).
- [Rui+18] J.E. Ruiz **and** others. ?Optimization of digital image processing to determine quantum dots’ height and density from atomic force microscopy? **in***Ultramicroscopy*: 184 (2018), **pages** 234–241. DOI: [10.1016/j.ultramic.2017.09.004](https://doi.org/10.1016/j.ultramic.2017.09.004).
- [SB21] R. Srikanth **and** K. Bikshalu. ?Multilevel thresholding image segmentation based on energy curve with harmony Search Algorithm? **in***Ain Shams Engineering Journal*: 12.1 (2021), **pages** 1–20. DOI: [10.1016/j.asej.2020.09.003](https://doi.org/10.1016/j.asej.2020.09.003).

- [SM00] Jianbo Shi **and** J. Malik. ?Normalized cuts and image segmentation? **in***IEEE Transactions on Pattern Analysis and Machine Intelligence*: 22.8 (2000), **pages** 888–905. DOI: [10.1109/34.868688](https://doi.org/10.1109/34.868688).
- [SMP11] Indrajit Saha, Ujjwal Maulik **and** Dariusz Plewczynski. ?A new multi-objective technique for differential fuzzy clustering? **in***Applied Soft Computing*: 11.2 (2011), **pages** 2765–2776. DOI: [10.1016/j.asoc.2010.11.007](https://doi.org/10.1016/j.asoc.2010.11.007).
- [Tao+20] Ran Tao **and others**. ?Analysis of Mafic rocks Microstructure damage and failure Process under Compression Test Using Quantitative Scanning Electron Microscopy and Digital Images Processing? **in***Engineering Fracture Mechanics*: 231 (2020), **page** 107019. DOI: [10.1016/j.engfracmech.2020.107019](https://doi.org/10.1016/j.engfracmech.2020.107019).
- [TKR20] Leslie Ching Ow Tiong, Seong Tae Kim **and** Yong Man Ro. ?Multimodal facial biometrics recognition: Dual-stream convolutional neural networks with multi-feature fusion layers? **in***Image and Vision Computing*: 102 (2020), **page** 103977. DOI: [10.1016/j.imavis.2020.103977](https://doi.org/10.1016/j.imavis.2020.103977).
- [UH05] R. Unnikrishnan **and** M. Hebert. ?Measures of Similarity? **in***2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1: IEEE, 2005*. DOI: [10.1109/acvmot.2005.71](https://doi.org/10.1109/acvmot.2005.71).
- [VKR08] Sara Vicente, Vladimir Kolmogorov **and** Carsten Rother. ?Graph cut based image segmentation with connectivity priors? **in***2008 IEEE Conference on Computer Vision and Pattern Recognition: IEEE, 2008*. DOI: [10.1109/cvpr.2008.4587440](https://doi.org/10.1109/cvpr.2008.4587440).
- [Wan+20] Xuan Wang **and others**. ?Leaf Recognition Based on Elliptical Half Gabor and Maximum Gap Local Line Direction Pattern? **in***IEEE Access*: 8 (2020), **pages** 39175–39183. DOI: [10.1109/access.2020.2976117](https://doi.org/10.1109/access.2020.2976117).
- [WJW09] Jie Wang, Lili Ju **and** Xiaoqiang Wang. ?An Edge-Weighted Centroidal Voronoi Tessellation Model for Image Segmentation? **in***IEEE Transactions on Image Processing*: 18.8 (2009), **pages** 1844–1858. DOI: [10.1109/tip.2009.2021087](https://doi.org/10.1109/tip.2009.2021087).
- [WO10] Hongzhi Wang **and** John Oliensis. ?Generalizing edge detection to contour detection for image segmentation? **in***Computer Vision and Image Understanding*: 114.7 (2010), **pages** 731–744. DOI: [10.1016/j.cviu.2010.02.001](https://doi.org/10.1016/j.cviu.2010.02.001).

- [XB13] Fengying Xie **and** Alan C. Bovik. ?Automatic segmentation of dermoscopy images using self-generating neural networks seeded by genetic algorithm? **in***Pattern Recognition*: 46.3 (2013), **pages** 1012–1019. DOI: [10.1016/j.patcog.2012.08.012](https://doi.org/10.1016/j.patcog.2012.08.012).
- [Xia+18] Min Xian **and others**. ?Automatic breast ultrasound image segmentation: A survey? **in***Pattern Recognition*: 79 (2018), **pages** 340–355. DOI: [10.1016/j.patcog.2018.02.012](https://doi.org/10.1016/j.patcog.2018.02.012).
- [ZA15] Nida M. Zaitoun **and** Musbah J. Aqel. ?Survey on Image Segmentation Techniques? **in***Procedia Computer Science*: 65 (2015), **pages** 797–806. DOI: [10.1016/j.procs.2015.09.027](https://doi.org/10.1016/j.procs.2015.09.027).
- [Zuc76] Steven W. Zucker. ?Region growing: Childhood and adolescence? **in***Computer Graphics and Image Processing*: 5.3 (1976), **pages** 382–399. DOI: [10.1016/S0146-664X\(76\)80014-7](https://doi.org/10.1016/S0146-664X(76)80014-7).
- [ZWY17] Shanwen Zhang, Xiaowei Wu **and** Zhuhong You. ?Jaccard distance based weighted sparse representation for coarse-to-fine plant species recognition? **in***PLOS ONE*: 12.6 (2017). **by editor** Zhao Zhang, e0178317. DOI: [10.1371/journal.pone.0178317](https://doi.org/10.1371/journal.pone.0178317).