Universidade Estadual de Londrina

GABRIEL JONAS AGUIAR

A META-LEARNING APPROACH FOR SELECTING IMAGE SEGMENTATION ALGORITHM

LONDRINA
2020

GABRIEL JONAS AGUIAR

A META-LEARNING APPROACH FOR SELECTING IMAGE SEGMENTATION ALGORITHM

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Supervisor: Prof. Dr. Sylvio Barbon Jr.

LONDRINA
2020

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

AG282 Aguiar, Gabriel.
A meta-learning approach for selecting image segmentation algorithm / Gabriel Aguiar. - Londrina, 2020.
59 f. : il.

Orientador: Sylvio Barbon.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2020.
Inclui bibliografia.

1. Meta-Learning - Tese. 2. Image Segmentation - Tese. 3. Machine Learning - Tese. I. Barbon, Sylvio. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519

GABRIEL JONAS AGUIAR

A META-LEARNING APPROACH FOR SELECTING IMAGE SEGMENTATION ALGORITHM

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Sylvio Barbon Jr.
Universidade Estadual de Londrina

Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina -
Departamento de Computação – UEL

Prof. Dr. Luiz Fernando Carvalho
Universidade Tecnológica Federal do
Paraná, Campus Apucarana – UTFPR

Londrina, 19 de março de 2020.

ACKNOWLEDGEMENTS

Primeiramente gostaria de agradecer a Deus por ter me capacitado nessa caminhada. Também agradeço aos meus pais, Valdeci e Elda, que me deram todo suporte e inúmeros motivos para agradecer. Gostaria de agradecer também ao meu irmão, Vinicius, que mesmo de longe me incentivou ao caminho da pesquisa.

Agradeço também aos meus amigos que sempre estiveram comigo e fizeram essa caminhada mais divertida e leve.

Agradeço ao meu orientador, professor e amigo, Sylvio Barbon, por todo apoio e orientação ao longo do mestrado e da graduação.

Agradeço também aos meus amigos do laboratório REMID, que sempre contribuíram com minha pesquisa.

Por fim, agradeço pelo apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"There isn't time, so brief is life, for bickerings, apologies, heartburnings, callings to account. There is only time for loving, and but an instant, so to speak, for that." - Mark Twain

AGUIAR, G. J.. **Seleção de algoritmos de segmentação baseado em meta-aprendizado**. 2020. 59f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2020.

RESUMO

Sistemas de Visão Computacional são muito utilizados e importantes em diversas aplicações. Dentro desses sistemas, o processo de detectação de uma região de interesse em imagens é de suma importância. Esse processo é conhecido como segmentação. Novos algoritmos de segmentação tem sido propostos nos últimos anos, porém nenhum é ótimo para qualquer imagem. Os métodos, usualmente, utilizados para seleção do melhor algoritmo são testar todos os algoritmos ou utilizar algum conhecimento passado sobre o problema. *Meta-learning* tem sido utilizado pela comunidade de pesquisa em *Machine Learning* para a recomendação do melhor algoritmo de *Machine Learning* para uma nova base de dados. Neste trabalho é investigado a hipótese que *Meta-Learning* também pode ser utilizado para recomendação do algoritmo de segmentação mais adequado. Os experimentos foram conduzidos com oito algoritmos de segmentação, com diferentes abordagens, baseadas em custo e complexidade computacional em uma base de *benchmark* composta de 300 imagens originais e 2100 imagens criadas por *augmentation*. Os resultados mostraram que através do *Meta-Learning* é possível recomendar o algoritmo de segmentação mais adequado com mais de 80% de acurácia para um grupo de algoritmos e com 69% para o outro grupo, superando os *baselines* considerando performance preditiva e de segmentação.

Palavras-chave: Meta-Learning. Segmentação de Imagem. Recomendação de algoritmo.

AGUIAR, G. J.. **A meta-learning approach for selecting image segmentation algorithm**. 2020. 59p. Master's Thesis (Master of Science in Computer Science) – State University of Londrina, Londrina, 2020.

ABSTRACT

Computer Vision Systems are used in many important real-life applications nowadays. Image segmentation is a key issue in Computer Vision Systems. New image segmentation algorithms have been proposed in recent years. However, there is no optimal algorithm for every image processing task. The selection of the most suitable algorithm usually occurs by testing every possible algorithm or using knowledge from previous problems. These processes can have a high computational cost. Meta-learning has been successfully used in the machine learning research community for the recommendation of the most suitable machine learning algorithm for a new dataset. We believe that meta-learning can also be useful to select the most suitable image segmentation algorithm. This hypothesis is investigated in this work. For such, we perform experiments with eight segmentation algorithms from two approaches, with different complexity and computational cost, using a segmentation benchmark of 300 images and 2100 augmented images. The experimental results showed that meta-learning can recommend the most suitable segmentation algorithm with more than 80% of accuracy for one group of algorithms and with 69% for the other group, outperforming the baselines used regarding recommendation and segmentation performance.

Keywords: Meta-Learning. Image Segmentation. Algorithm recommendation.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANN Approximated Nearest Neighbour.

BG Brightness Gradient.

BGTG Brightness/Texture Gradient.

BSDB500 Berkeley Segmentation Dataset and Benchmark.

CD Critical Difference.

CG Colour Gradient.

CGTG Colour/Texture Gradient.

CVS Computer Vision System.

DS Decision Stump.

DT Decision Tree.

EME Measure of Enhancement.

FFT Fast Fourier Transform.

GBM Gradient Boosting Machine.

GLCM Gray Level Co-Occurrence Matrix.

gPb Global Probability of Boundary.

gPB-ucm Global Probability of Boundary Ultrametric Contour Maps.

IG Information Gain.

IQA Image Quality Assessment.

LBP Local Binary Patterns.

LOO-CV Leave-One-Out Cross Validation.

ML Machine Learning.

MtL Meta-Learning.

OOB Out-of-Bag.

OWT Oriented Watershed Transformation.

PSO Particle Swarm Optimization.

RBF Radial Basis Function.

RF Random Forest.

SNM Statistical Naturalness Measure.

SPC Sparse Code Gradient.

SVM Support Vector Machine.

TG Texture Gradient.

UCM Ultrametric Contour Maps.

XGBoost Extreme Gradient Boosting.

CONTENTS

1 INTRODUCTION

Computer Vision is a field of research which is an intersection area between the Image Processing and Artificial Intelligence, and studies techniques, methods and hardware to emulate the human vision in real-life applications [1]. Computer Vision has been applied in current problems in many areas, such as medical applications [2, 3], food quality evaluation [4, 5, 6], and self-driving cars [7].

In Computer Vision System (CVS), one of the main steps is to detect the region of interest or its boundaries and separates it from the non-interest region. This part of the CVS is known as image segmentation.

Also, image segmentation is one of the most studied problems in computer science [8, 9]. It is one of the most challenging tasks to be automatically performed [10, 11, 12, 13]. According to Fernandez *et al.* [8], segmentation is usually a step in an image processing chain or pipeline. When the image segmentation step is inadequate, it harms the performance of the whole pipeline. Although additional operations can reduce the drawbacks of a poor quality segmentation, they will increase the pipeline size and complexity [14]. Therefore, the selection of the most suitable segmentation algorithm can improve the image processing performance as a whole.

Additionally, adopting the *No free lunch* theorem often used in Machine Learning (ML) [15], no segmentation algorithm will be the most suitable for all images. The author believe that look for the most suitable segmentation algorithm for each image would be the best alternative. However, this alternative has a high computational cost. A similar option with lower cost would be to recommend, if not the best, at least a suitable segmentation algorithm for each new image. This study proposes and experimentally investigates a Machine Learning (ML) based recommender system to deal with this task [16, 17, 11].

A similar problem occurs when recommending the most suitable ML algorithm for a new dataset. This problem has been successfully investigated using Meta-Learning (MtL) [18, 19]. MtL is a ML approach to learn from previous ML experiences. For such, each algorithm induces a meta-model able to map the description of a dataset by a set of meta-features to the performance of a set of ML algorithms when applied to this dataset.

Meta-Learning has been successfully employed to select the best algorithm for data stream problems [20], ranking a set of algorithms [21], for a new dataset. Also, MtL has been applied for optimization of hyperparameters of learning algorithms [22, 23, 24].

In image analysis, MtL has also been applied to image segmentation [25, 11], to object detection and localization [26], and to search similar images [27]. Campos *et al.* [11] addressed the algorithm recommendation problem for image segmentation. The authors

used MtL to recommend one among three conventional image segmentation algorithms for three restricted scenarios [28, 29, 30]. Their study did not include the most recent image segmentation approaches [31, 32, 33, 34]. Furthermore, the authors employed hundreds of images from each image domain. However, in most real-life image segmentation, it is only possible to deal with a few images.

In this study, we investigate the use of MtL to recommend segmentation algorithms for a new unseen image. For such, we describe each image by a set of meta-features extracted from itself. These meta-features should provide the necessary information to allow the recommendation of suitable segmentation algorithms.

In order to deal with trade-offs associated with the automatic image processing system, segmentation performance and computational complexity, the algorithms used in this work were separated into two groups: Gradient-based [34] and Machine Learning-based [31, 32, 33]. The Gradient-based group of algorithms are low-cost algorithms, based on simple image features, such as color, texture and brightness. On the other hand, the Machine Learning-based algorithms are more computationally expensive and based on learning algorithms. Naturally, the segmentation performance of ML-based algorithms is better; however, the computational cost makes it difficult to use it in limited hardware. A brief summary of our proposed approach is presented in Figure 1. Given an input image, the system is going to predict which is going to be the most suitable algorithm from each group.

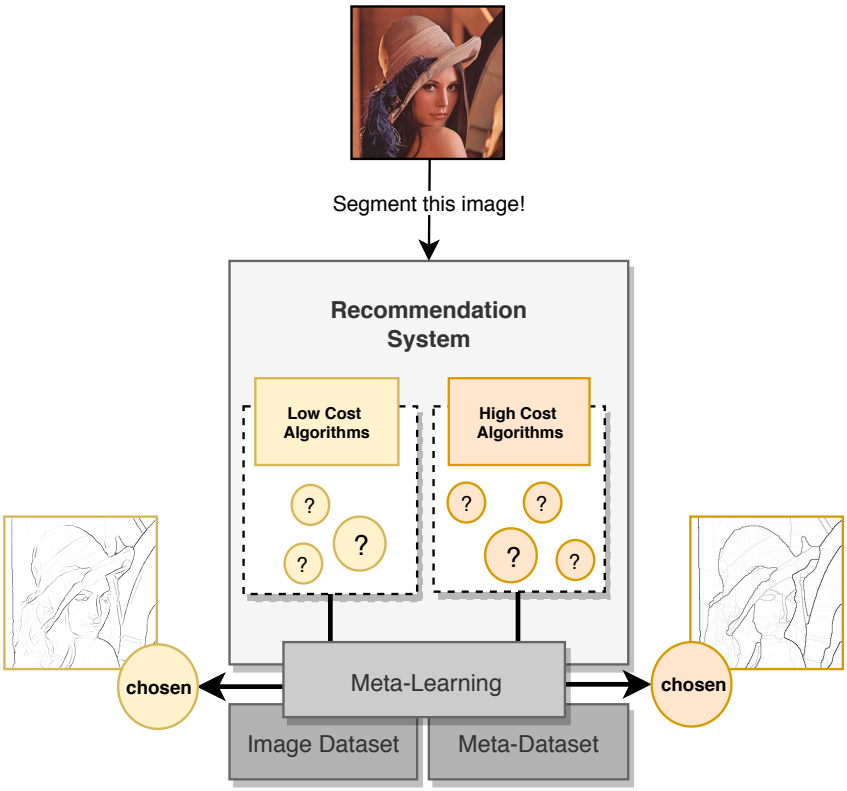


Figure 1 – Graphical summary of the proposed recommendation system.

The experiments were conducted using a well-known database of segmentation benchmarking, composed of different scenarios and domains, the Berkeley Segmentation Dataset and Benchmark (BSDB500). This dataset was already used in several tasks, particularly for the evaluation of new segmentation algorithms [32, 35, 36, 37]. The BSDB500 also contains the ground-truth segmented images, making it possible to evaluate the segmentation algorithms without bias and objectively.

In our experiments, three supervised classification algorithms, Random Forest (RF), Support Vector Machine (SVM) and Extreme Gradient Boosting (XGBoost), were evaluated as meta-learners. Their predictive performances were compared with two different baselines: (i) Not use the recommendation system and always choose the same segmentation algorithm (`OneR`) and (ii) randomly choose the segmentation algorithm (`Random`).

1.1 Objectives and contributions

The main objective of our work is to propose a recommending system for selecting the most suitable segmentation algorithms based on Meta-Learning for a given image.

The main contributions of this work can be summarized as:

1. The use of Meta-Learning to select segmentation algorithms using a dataset composed with images from different scenarios, and to state the superiority of the recommendation system regarding the baselines;
2. Explore two different groups of segmentation algorithms (Gradient and Machine Learning-based) based on computational complexity and segmentation performance;
3. Investigates new meta-features for image related problems;

1.2 Outline

This master thesis is structured as follows. Chapter 2 presents, as related work, a revision of works that applied MtL in the Computer Vision area. Chapter 3 presents a theorical foundation about Image Segmentation, Machine Learning and Meta-Learning. The materials and methods are covered on Chapter 4. Experimental results are presented and analysed in Chapter 5. Finally, main conclusions and future work directions are discussed in Chapter 6.

2 RELATED WORK

Meta-Learning (MtL) has been successfully used for algorithm selection [38]. In the recent years, it has been applied to solve several problems in Computer Vision. In Yong *et al.* [16], the authors proposed a learning-based approach to select image segmentation algorithms. The approach was explored with SVM for modelling a recommender based on 9000 images. The recommender was compared with the manual selection of thresholding algorithms (Otsu, enhanced histogram, region growing and Kapur's maximum entropy). Although the results presented a high predictive performance, only grayscale synthetic images, described by their histogram values, were used in the experiments. Thus, the contributions were limited to specific scenarios using simple image segmentation algorithms, with low applicability to real-life images.

A novel framework for intracellular image segmentation based on effective algorithm selection was proposed by Takemoto and Yokota [17]. When recommending an algorithm for a new image, the system compares the segmented regions with user-supervised regions based on similarity measures. The authors' framework was able to recommend suitable algorithms using intensity and texture meta-features (pixel intensity and normalised moment), as well as to define the hyperparameter of each segmentation algorithm. Although the authors presented a segmentation solution based on SVM and Approximated Nearest Neighbour (ANN), the contribution was limited to a very specific domain. They explored just two image features with two types of hyperparameter settings.

In Attig and Perner [25], MtL was used to identify the best set of hyperparameter values to perform watershed segmentation. The authors used hierarchical clustering to identify similar images. They used only nine images of different types (biological images, faces, and animals). If an image was considered similar to a well-segmented image, it was segmented using the same hyperparameter values used for that image. The authors used meta-features from image descriptions (statistical and texture features). In order to improve performance, they computed particular weight values to the statistical and texture features. In our work, we explored the image feature importance to understand the meta-features contribution and use this information to automate the segmentation recommendation task.

MtL was successfully used to select segmentation algorithms in Campos *et al.* [11]. In the experiments, the authors applied 3 segmentation algorithms to 4 image datasets. According to the results, MtL was able to select suitable segmentation algorithms. However, the datasets were limited to restricted domains, which does not allow a generalisation to other image domains. Besides, the segmentation evaluation was subjective, provided by domain experts, increasing the cost of the framework.

Al Marakeby *et al.* [26] proposed a framework to generalised object detection and localisation based on MtL. The authors reported AdaBoost as the most promising meta-learner, when comparing with Decision Stump (DS) and SVMs. Their framework did not include hyperparameter tuning. The experiment was run on three different datasets, with a total of 2242 images. Despite the high performance and the identification of the most relevant meta-features, the proposed framework presented a high computational cost. Besides, the experiments were restricted to only three application domains.

A variant of the Particle Swarm Optimization (PSO) meta-heuristic using MtL was proposed by Yamaguchi *et al.* [27]. The authors investigated the problem of searching for satellite images by similarity in a dynamically changed problem space. They took advantage of transfer learning, a MtL technique that transfer knowledge acquired in a dataset to model another dataset. The obtained results were explored in a ML perspective of a single application domain (satellite imaging), without a general contribution to image processing. Different from the other works, the authors explored shape descriptors as meta-features.

It is possible to identify relevant contributions in different domains for image segmentation algorithm recommendation. However, most of the proposed approaches are restricted to a single application domain. Further, they do not explore new segmentation algorithms. Table 1 summarises the main aspects of the related works highlighting important comparison items. Finally, the importance of image features was not taken into account. In this work, we investigate the use of MtL for image segmentation algorithm recommendation using novel image segmentation algorithms. We also use image features in the recommendation process and perform experiments with images from several different domains.

Table 1 – Summary of related studies applying recommendation for image segmentation tasks. Fields without information in the related study are marked with a hyphen.

Task	**# of algorithms**	**# of images**	**Image features**	**Image domains**	**Evaluation criteria**	**Feature analysis**	**Reference**
Segmentation Algorithm Selection	4	9000	Histogram	Synthetic	Manual	-	[16]
	2	6	Intensity, Boundary	Intracellular	Automatic	-	[17]
	3	366	Color, Intensity, Texture, Histogram, Image quality	Chicken, Cloud, Wound	Manual	•	[39]
Hyperparameter recommendation	1	9	Statistical, Texture	Biological, Faces, Animals	Manual	•	[25]

3 THEORETICAL FOUNDATION

In this chapter, the main concepts explored in this thesis are presented. In Section 3.1, an overview of the Computer Vision concepts is provided. In this same section, Image Segmentation (Section 3.1.1), topics more specifically related to this work are reviewed. Section 3.2 presents the concepts of Machine Learning and the ML algorithms related to this work. Moreover, since we are dealing with an algorithm selection task, and using Meta-Learning (MtL) to solve it, the definition of MtL is introduced in Section 3.3.

3.1 Computer Vision

Computer Vision is an area of research, which studies techniques and methods to use computers to emulate human vision and also be able to learn and take actions based on visual inputs [1]. These techniques and methods are used to build a Computer Vision System (CVS).

Computer Vision Systems (CVSs) are widely used in real-life scenarios from different domains and various types of environments. CVS were used to evaluate chicken quality [4], predict papaya maturity stage [40] and to improve quality inspection in food products [41]. Also, it was used for vehicle tracking and traffic surveillance [42], plant species identifications [43].

According to Umbaugh [44], a Computer Vision System is commonly composed of some tools: (i) Image Segmentation, (ii) Feature Extraction, (iii) Image Transforms and (iv) Pattern Classification. Image Segmentation is used to find objects from the raw image. To get information about the image, such as shape, color, intensity, the process of Feature Extraction is used. Also, acquiring information from different domains (frequency, time-frequency) is essential, then, image transforms may be used. The final step is to make decision about the input data after being processed.

In this study, the main focus is on the Image Segmentation step. Since it is one of the main points of a Computer Vision System, to select the most suitable algorithm is essential for the success of the CVS application. The concepts about Image Segmentation and the segmentation algorithms used in those works are presented in this section.

3.1.1 Image Segmentation

Image segmentation is one of the significant steps in a Computer Vision System (CVS). Generally, it can be defined as a process to identify some region of interest in the image [1]. Also, segmentation may be seen as a method to subdivide an image into regions or objects. The segmentation of nontrivial images is one of the most challenging

tasks in digital image processing [1], and its precision is directly related to the success of the CVS. Therefore, image segmentation is one of the most studied problems in Image Analysis and Computer Vision [8].

Many image segmentation algorithms have been proposed in the literature [1, 45]. These algorithms are based on different methods: thresholding, region growing, clustering, supervised classification, edge detection and others.

In this work, 8 segmentation algorithms based on edge detection and supervised classification were used. The selection criteria of the algorithms was the Berkeley Segmentation Dataset and Benchmark ranking of algorithms. They were divided into two groups based on their computational complexities. The first group is composed of 5 algorithms of edge detection with low computation complexity based on Color, Texture and Brightness Gradient. In the other group, 3 algorithms were selected. They also are based on edge detection; however, supervised classification is used, which improves the precision and also increases their computational cost. In the following subsections, both groups and their algorithms are presented.

It is worth to emphasize that the methods, hyperparameters and experimental choices presented in the following subsections are proposed by the authors of the segmentation algorithms. The implementation of the algorithms are available on [1].

3.1.1.1 Gradient-based

Given a pair of pixels, imagine a line connecting them in the image plane. If the pixels are in different segments, we expect to find a discontinuity indicating a large gradient along the line. Then, we apply a operator to measure the degree to which some descriptor varies at a pixel (x, y) in direction θ, in the nearby pixels. In this sense, we computed the gradient in three image descriptors: Color, Brightness and Texture.

For all algorithms in this group, the idea is to create a map of probability, i.e., an image, in which the intensity of each pixel is the probability of the pixel at the same position in the input image be part of the boundary.

To create this map, for each pixel (x, y) in the input image, its neighbours within a radius r are selected and halved with an orientation θ. Then, a histogram is extracted from both halves. The histogram extraction is what distinguish the Gradient-based algorithms, their details are discussed later in this section. Subsequently, the histograms of each half are compared. In order to evaluate the histograms in different positions and directions, it is computed using 8 values of θ, in the interval $[0, \pi)$ equally divided. Also, when two types of gradient are used in combination, their histogram difference is calculated independently and then combined for the final part. Finally, the probability of that pixel be a part of

[1] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html

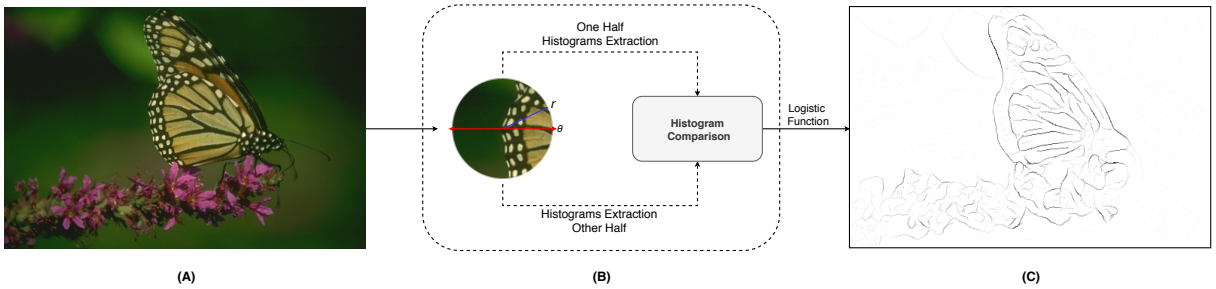


Figure 2 – Summarizing the process to segment a image using the gradient based approach. **(A)** Image input. **(B)** Gradient computation process. It is applied for each pixel with different values of θ. **(C)** Ouput Image (Map of probability).

the boundary is obtained by applying the values of gradient resulting from histogram comparison in a logistic function. Figure 2 illustrates this process.

In this work, all the algorithms were implemented in MATLAB, using the implementation of Berkeley Segmentation Dataset and Benchmark [2] relying on their default hyperparameters.

3.1.1.1.1 Color and Brightness Gradient

The segmentation task for a human is usually an easy task since with our visual perception, it is trivial to identify different objects and their boundaries. Then, to extract the Color and Brightness with respect to human perception, the CIELAB color space [46] was used. The CIELAB color space express color as three values: L^* for the lightness from black to white, a^* from green to red, and b^* from blue to yellow.

The extraction of both types of Gradients is similar. The main difference is the input. The Brightness Gradient is extracted using the L^* values, the a^* and b^* are used to compute the Color Gradient.

The used approach [34] is based on binning kernel density estimates of the CIELAB using a Gaussian kernel and comparing the histograms. Firstly, the image is divided into N bins, according to similar values. At the same time, a $N \times N$ color similarity matrix is estimated using a Gaussian function. Then, for each bin, one binary image is created (1 in that bin, 0 for not in that bin). After that, in each binary image, every pixel is evaluated using its neighborhood. The value of the pixel is going to represent how homogeneous the neighbourhood region is. The neighborhood is defined by a radius r, divided into two halves with an orientation θ. The halves are evaluated individually, and then the absolute difference between the halves is calculated. Finally, the resulting matrix, which contains every homogeneity from each region, is multiplied by the color similarity matrix. The resulting matrix is the Color or Brightness Gradient.

[2] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

3.1.1.1.2 Texture Gradient

The Texture gradient is an operator that measures the degree to which texture of scale r varies at a pixel (x, y) in direction θ. Similar to the Brightness and Color gradient, the texture dissimilarity is computed in the two halves of a circle centered at a point and halved.

For each pixel, it is associated an array of 13 filter responses centered at the pixel. The filters have properties to highlight boundaries. Then, each half of the disc contains a set of filter response arrays that we can visualize it as a cloud of points in a space with dimensionality equal to the number of filters.

Therefore, it is necessary to compare the halves to get the value of the Texture gradient. In our work, it was used the texton approach. It estimates the joint distribution of filter responses using adaptive bins. The responses arrays are clustered using k-means. The clusters centers define texture primitives (textons), which are linear combinations of the filters. It is worth to mention that the textons were previously computed using the training set of the BSDB500.

Once the textons have been computed, each pixel is assigned to the nearest texton. Then, the texture difference is extracted by comparing the histogram of texton bins in the two halves using χ^2 (Equation 3.1), where g and h are the histograms and i is the respective bin.

$$\chi^2(g, h) = \frac{1}{2} \sum \frac{(g_i - h_i)^2}{g_i + h_i} \tag{3.1}$$

3.1.1.2 Machine Learning-based

To improve the generalization and the boundary detection despite the computational cost, some authors proposed new boundary detectors using Machine Learning (ML) combined with the Gradient-based approach. In this work, three ML-Based algorithms were selected: Global Probability of Boundary (gPb), Global Probability of Boundary Ultrametric Contour Maps (gPB-ucm) and Sparse Code Gradient (SPC).

3.1.1.2.1 Global Probability of Boundary

There are many approaches in the literature on contour detection [33]. The main idea of the Global Probability of Boundary (gPb) is to combine two approaches: Local information and Global information.

Local approaches aim to quantify the presence of a boundary using local measurements. Gradient-based algorithms (Section 3.1.1.1) are an example of segmentation

algorithms which uses local information to detect boundaries. The global approach relies on integrating information into the grouping process.

In the gPb, the Gradient-based approach is used for local information detection. Considering $Pb_r(x, y, \theta)$ the function which predicts the probability of each pixel to be a boundary, by measuring the difference in some feature channels on the disc of radius r centered at (x, y) and divided in half with an orientation θ. Color, Brightness and Texture gradients were used at three different scales: $[\frac{r}{2}, r, 2r]$ where r is the default radius of Pb detector. The gradients are linearly combined, named as G_i, in a single multi-oriented signal:

$$mPb(x, y, \theta) = \sum_{i=1}^{9} \alpha_i \cdot G_i(x, y, \theta) \tag{3.2}$$

To add global information, first we need to solve the generalized eigenvectors of the linear system (Equation 3.3), where W is an affinity matrix which entries represent the similarity between pixels, and D is defined as $D_{ii} = \sum_j W_{ij}$.

$$(D - W)v = \lambda Dv \tag{3.3}$$

Then, we consider the first $k+1$ generalized eigenvectors $(v_0, \ldots, v_k)$ of the system, with the corresponding eigenvalues $\lambda_0, \ldots, \lambda_k$. In the implementation used in this work, it was used $k = 8$.

The affinity matrix W is constructed using the intervening contour cue [47], the maximal value of mPb along a line connecting two points. Then, we use Gaussian directional derivatives at multiple orientations θ to extract the contours from the eigenvector v_j, obtaining an oriented signal $sPb_{v_j}(x, y, \theta)$. Given that we have multiple eigenvectors, their information is combined, as in Equation 3.4, to be used as a component of the boundary detector.

$$sPb(x, y, \theta) = \sum_{j=1}^{k} \frac{1}{\sqrt{\lambda_j}} \cdot sPb_{v_j}(x, y, \theta) \tag{3.4}$$

Thus, we have local information from the mPb detector and global information from the sPb signal. A simple linear combination of the signal is used as the final detector. Therefore, the Global Probability of Boundary is defined in Equation 3.5. The weights α_i and γ are learned by gradient ascent using the training set of the Berkeley Segmentation Dataset and Benchmark. The real-valued contour image is obtained from gPb by

considering the maximal response over orientations at each location.

$$gPb(x, y, \theta) = \sum_{i=1}^{9} \alpha_i \cdot G_i(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \tag{3.5}$$

3.1.1.2.2 Global Probability of Boundary Ultrametric Contour Maps

Given an output of the Global Probability of Boundary (gPb) detector there are some problems with the contours because often they are not closed and therefore do not partition the image into regions. Then, Arbelaez *et al.* [32] proposed improvement for gPb, the Global Probability of Boundary Ultrametric Contour Maps (gPB-ucm). The main idea is to use the gPb as input and apply hierarchical segmentation to endorse the contours.

First, using the gPb output, the maximum value of the contour detector over each orientation is taken by computing $E(x, y) = max_\theta\{gPb(x, y, \theta)\}$. Then, the local minimal points are used as seed locations for homogeneous segments, and the watershed transformation is applied [48]. After the watershed transformation, the image is divided into N regions, but the boundaries are not real-valued, and it is not possible to use the mean value of $E(x, y)$, because boundaries that were supposed to be weak may be upweighted due to the pixel orientation and the final boundaries are going to be compromised.

To assign real-values to the boundaries, the value of each pixel is assigned using an approximation of their orientation. The orientation of each pixel is done by approximating the border of the watershed defined regions with line segments. Then, for each pixel, it is associated with an orientation $o(x, y) \in [0, \pi)$. Therefore, the boundary probability value is the mean of $E(x, y, o(x, y))$. This process is named as Oriented Watershed Transformation (OWT). Hence, we have an image that each border of the watershed transformation is associated with a real-value representing the boundary probability.

After the OWT process, the image is segmented using hierarchical segmentation. The core idea of hierarchical segmentation in the boundary map is to separate strong boundaries and weak boundaries. In the higher levels of the hierarchy, there are only strong contours, resulting in an undersegmenation, while in the base level, weak contours are taken into account, resulting in an oversegmentation. Moving through the levels is a trade-off between these extremes.

The hierarchical segmentation process applied in this algorithm is the Ultrametric Contour Maps (UCM). The algorithm proceeds by sorting the links by similarity and merging similar regions. Based on the initial points and the regions of the OWT, and a matrix W which computes the dissimilarity between the regions, it creates a tree, in which the root is the entire image and the leaves are the initial points. The dissimilarity matrix is computed using the mean of the pixels in the common boundary of two regions.

Finally, the output is the tree, and we define in which level k we want to segment. In our experiments, the k value was the default that was obtained using the training set of the Berkeley Segmentation Dataset and Benchmark.

3.1.1.2.3 Sparse Code Gradients

In the Global Probability of Boundary (gPb), the main idea is to use combined hand-designed local features (Gradients) and aggregate with global information in order to find the boundaries. However, instead of using Gradients as local features, Xiaofeng and Bo [31] replaced it with Sparse Code Gradient (SPC), which is a rich representation automatically learned from data. The SPC is more effective in capturing local contour information than the Gradients previously proposed. When applied to the Berkeley Segmentation Dataset and Benchmark validation set, the SPC approach achieved the best results among all the evaluated segmentation algorithms.

Firstly, we need to extract the sparse code for every pixel. In order to do that, it is used the K-SVD [49] and the Orthogonal Matching Pursuit [50]. The K-SVD is a generalization of the K-Means and learns dictionaries of codewords from unsupervised data. Given a set of image patches $Y = [y_1, \ldots, y_n]$, K-SVD finds a dictionary D and an associated sparse code matrix X by minimizing the Equation 3.6, in which $\|\cdot\|_F$ denotes Frobenius Norm, x_i the columns of X, the zero-norm $\|\cdot\|_0$ counts the non-zero entries in x_i, and K is a previously defined value of tolerance for non-zero entries, defining the sparsity level. Given a fixed dictionary D, the matrix X is optimized using the Orthogonal Matching Pursuit (OMP), a greedy algorithm for finding sparses codes. With the optimized X, the dictionary D and its associated sparse coefficients are sequentially updated by singular value decomposition. At this point, we have dictionary D learned, and then, we can use the OMP to compute the sparse codes at every pixel. It is important to mention that, the learning step is done one time, using the training set of BSDB500, then we use the same dictionary for the other images. Also, the sparse codes are computed using the Chromacity channel ($a*$ and $b*$ of CIELAB color space).

$$\min_{X,D} \|Y - DX\|_F^2 \quad s.t.\ \forall i,\ \|x_i\|_0 \leq K \tag{3.6}$$

Once we have computed the sparse codes for every pixel, as in the gPb, the neighborhood of the pixel is evaluated. For each pixel p and scale s, two rectangles N^a and N^b of size s-by-$(2s+1)$ on both sides of p with orientation θ are used as the pixel neighborhood. Then, we represent each half using function F (Equation 3.7), an average pooling of non-zero entries and compute the difference between the halves with Equation 3.8. In G, the difference is normalized due to the variation of the sparse codes. It makes the features

robust to variations and increases their discriminative power.

$$F(N) = \left[\frac{\sum_{i \in N} |x_{i1}|}{\sum_{i \in N} I_{|x_{i1}|>0}}, \ldots, \frac{\sum_{i \in N} |x_{im}|}{\sum_{i \in N} I_{|x_{im}|>0}} \right] \tag{3.7}$$

$$G(N_s^a, N_s^b) = \frac{|F(N_s^a) - F(N_s^b)|}{\|F(N_s^a)\| + \|F(N_s^b)\| + \epsilon} \tag{3.8}$$

At this point, we could train a regressor for each scale s and use it as boundary probability. However, the authors of SPC find out that is better to use a multi-scale feature vector $\hat{X}$, in order to allow interactions between scales. Then, for each pixel p and scales $s \in 1, \ldots, S$:

$$\hat{X}_p = [G(N_1^a, N_1^b), \ldots, G(N_S^a, N_S^b); F(N_1^a \cup N_1^b), \ldots, F(N_{S1}^a \cup N_S^b)] \tag{3.9}$$

As $\hat{X}_p$ is high dimensional, it was used linear Support Vector Machine (SVM) as a regressor. However, learning linear function directly from $\hat{X}_p$ was not efficient, then it was applied a double power transformation on the feature vector, i.e., $\hat{X}'_p = [\hat{X}_p^{\alpha_1}, \hat{X}_p^{\alpha_2}]$, with $\alpha_1 < \alpha_2 < 1$, because it reshapes the distribution of the features into a more Gaussian form [51].

3.2 Machine Learning

Machine Learning (ML) [52] is an area of study of Artificial Intelligence. Given a set of examples $\mathcal{E}$ and their descriptors or features $\mathcal{X}$ and a target value $\mathcal{Y}$, the main goal of ML is to find a function f which maps $f : \mathcal{E} \times \mathcal{X} \rightarrow \mathcal{Y}$. The process to find the function f is named *training phase*. In the *training phase*, previous experience of the problem is used to adjust the function and try to minimize the error.

There are mainly two approaches of ML: (i) Supervised and (ii) Unsupervised. While in the supervised approach, the real values of the target $\mathcal{Y}$ are known, in the unsupervised they are not. In this work, the supervised approach is used, since we know which segmentation algorithm is better for each image.

Regarding the problems we can apply Supervised ML, the domain of the target $\mathcal{Y}$ is important to define which algorithm we are going to use. If the target is a continuous value, it is a Regression problem. In our work, the target is which algorithm from each group is better for the image. Therefore, our problem is a Classification problem since our target is discrete.

In this work, three ML algorithms, with different learning biases, were experimented: Random Forest (RF), Support Vector Machine (SVM) and Extreme Gradient

Boosting (XGBoost). These algorithms were selected due to their widespread use and capacity of high-performance models induction.

3.2.1 Random Forest

Random Forest (RF) [53] is an ensemble learning method. Ensemble methods combine multiple weak learning algorithms to obtain better predictive performance than could be obtained from the learning algorithm alone. RF combines multiple Decision Trees (DTs), trained on different subsets of the training data. Different subsets are built for each tree using random sampling with replacement. Also, each tree uses a subset of randomly chosen features.

Decision Trees are MLs algorithms that represent the extracted knowledge in a tree-based model, in which each node represents a selected feature of the dataset. The DT algorithm builds binary trees by recursively partitioning the data space into orthogonal hyper-planes [54]. Each node splits the data into smaller and more homogeneous subsets. To select which feature is going to be used to split the data, a dispersion measure like Gini Index or Information Gain is used. Therefore, the feature that splits the data into a more uniform subset is selected.

To compute the Information Gain of a selected feature $\hat{x}$, firstly we need to compute the entropy of each subset that would be created if we split the data using the feature $\hat{x}$. It is important to mention that if $\hat{x}$ is a discrete value, each value of $\hat{x}$ is used to split the data, otherwise, the mean value of $\hat{x}$ is used. Then, the entropy of subset $\mathcal{S} \subseteq \mathcal{D}$, where $\mathcal{D}$ is the original dataset, is computed as:

$$H(\mathcal{S}, \hat{x}) = -\sum_{i=1}^{Y} p_i \log_2 p_i \tag{3.10}$$

where p_i is the probability of a class, i appears in the subset $\mathcal{S}$, and Y is the total number of classes. Next, the weighted mean of the entropy of each subset is calculated:

$$H(\mathcal{D}, \hat{x}) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{S}_{\hat{x}}|} H(\mathcal{S}_i, \hat{x}) |\mathcal{S}_i| \tag{3.11}$$

where $|\cdot|$ denotes the size of the set, and $|\mathcal{S}_{\hat{x}}|$ represents the number of subsets after split the data with feature $\hat{x}$. Finally, the Information Gain (IG) is the difference between the general entropy of the data $H(\mathcal{D})$ and the weighted mean of the entropy of the subdivided data with the feature $\hat{x}$, therefore, $IG = H(\mathcal{D}) - H(\mathcal{D}, \hat{x})$. The IG is computed for every feature, and the one with the higher value is selected.

Thus, n decision trees are grown with n random subsets of the training set, with m features, also randomly selected from the feature space. Figure 3 presents a representation

of a RF. To predict the class of a sample, it works similarly to an election, each DT votes and the mode is the given prediction.

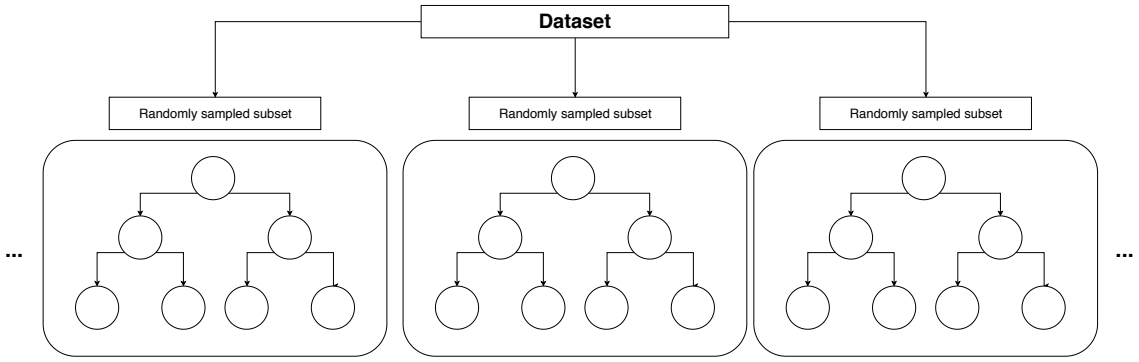


Figure 3 – Representation of a Random Forest.

3.2.2 Support Machine Vector

Support Vector Machines (SVMs) [55] are kernel-based algorithms that perform classification by finding the best hyperplane to separate the classes in the feature space and maximize the decision margin. The decision margin is the perpendicular distance between the separating hyperplane and the samples closest to the hyperplane. These samples are the support vectors. SVMs are known to be robust, handling a wide variety of problems, presenting high accuracy and capacity to treat high-dimensional data.

A hyperplane is a $(n-1)$-subspace for an n-dimensional space. It is defined as in Equation 3.12. Evidently, it can only separate classes that are linearly separable. In linearly separable problems, SVM tries to find a hyperplane that separates the classes, and optimize it to maximize the decision margin and correctly classify 100% of the examples. However, most of the times, the datasets are non-linearly separable. An example of a non-linearly separable problem is presented in Figure 4. To deal with non-linear problems, the SVM brings two concepts: **(i)** Soft Margin and **(ii)** Kernel Trick.

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \cdots + \beta_n * x_n = 0 \tag{3.12}$$

Soft Margin can be seen as a tolerance degree for misclassified samples. The idea is to balance the trade-off between finding a line that maximizes the margin and minimizes the misclassification. The degree of how much soft the margin will be is an important hyper-parameter in the SVM, and it is commonly denominated as C. The higher the C, the more penalty SVM gets when it makes misclassification, which leads to a tighter margin, and fewer support vectors in the decision margin. Figure 5 illustrates the impact of different C values in the decision margins.

Besides the Soft Margin, to perform hyperspace transformation based on kernels is the Kernel Trick. It maps the inputs into a high dimensional feature space where the

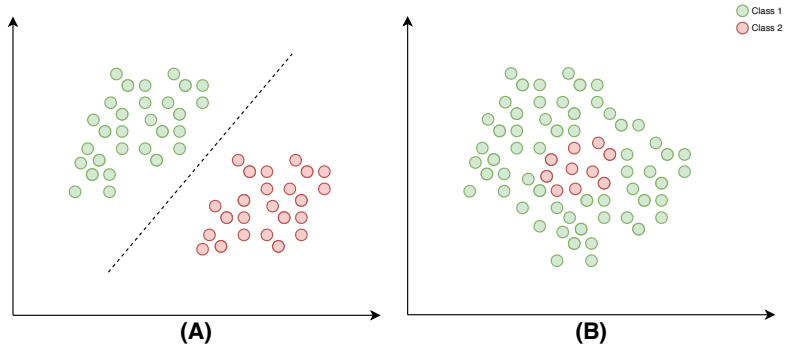
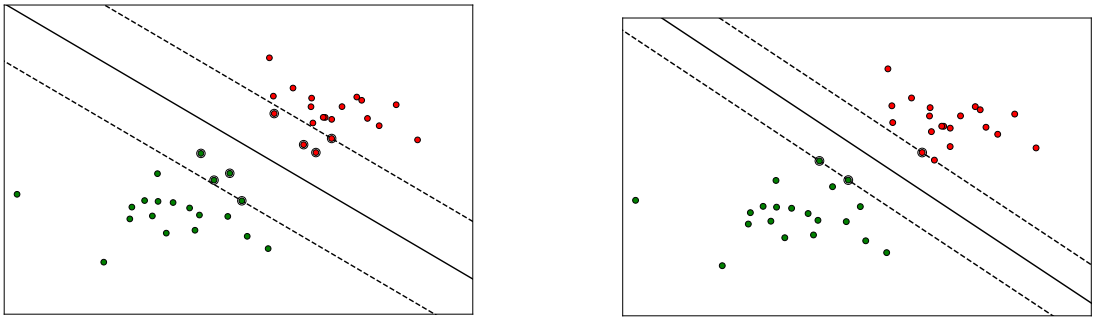


Figure 4 – Example of a **(A)** linearly and a **(B)** non-linearly separable problem.



(a) $C = 0.1$ (b) $C = 1.0$

Figure 5 – Tolerance degree (C) variation in a SVM with linear kernel.

problem is linearly separable, using kernel functions. The most popular kernels are **(i)** Polynomial and **(ii)** Radial Basis Function (RBF). When the Polynomial kernel is used, a polynomial combination is applied to all existing features to create new features. The Radial Basis Function (Equation 3.13) generates new features by measuring the distance between all samples to a specific sample. In RBF, the γ value is used to control the influence of new features in the decision process. The higher the γ, the more influence of new features in the decision boundary. In Figure 6, an example of non-separable data, which is mapped to a linearly separable feature space.

$$\phi(x, center) = e^{-\gamma||x-center||^2} \tag{3.13}$$

Support Vector Machine is an intrinsically binary classifier, since it is designed to separate two classes. However, it is possible to use SVMs in multiclass problems, such as our algorithm recommendation problem, by using the One-vs-All approach. For a problem with N classes, One-vs-all classification is a method which involves training N distinct binary classifiers, each designed for recognizing a particular class. Then those N classifiers are collectively used for multi-class classification. Figure 7 illustrates this method, each color indicates a class and each line is the separation margin for that class.

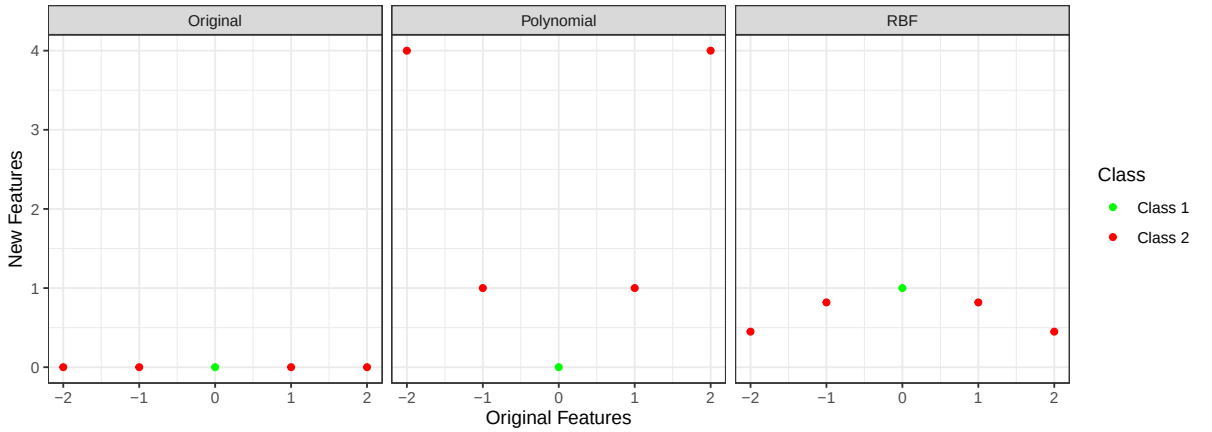


Figure 6 – Non linear hyperspace transformation example. Polynomial transformation with degree = 2 (x^2). In RBF, the center was $(0, 0)$ and $\gamma = 0.2$.

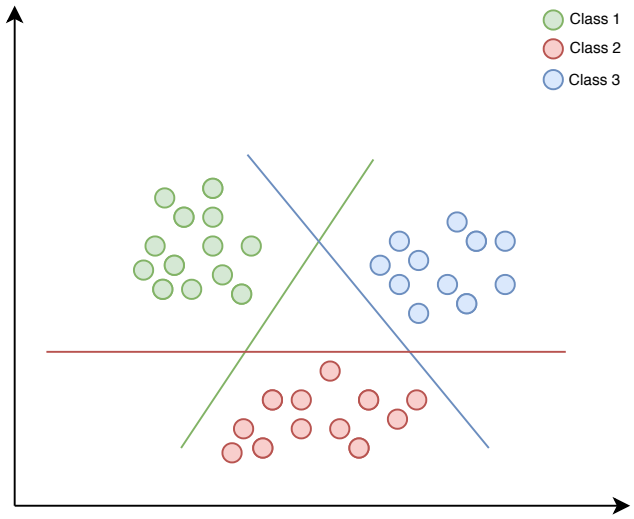


Figure 7 – One-vs-all approach illustration.

3.2.3 XGBoost

Extreme Gradient Boosting (XGBoost) [56] is a scalable end-to-end tree ensemble boosting machine learning system based on the Gradient Boosting Machine (GBM) [57] framework that has provided state-of-the-art results on many problems. In fact, XGBoost was reported as the best prediction technique in several data mining competitions. The XGBoost implementation adds new features to GBM in order to improve predictive performance and also the computational time.

Similar to RF, GBM is an ensemble of trees. However, in RF, the predictors are independent and combined using some average technique (average, majority voting). This method is known as Bagging. In GBM, the boosting method is used to create the ensemble. The trees are sequentially added to the ensemble aiming to minimize the current error gradient. In Figure 8, the difference between the methods is exemplified. The idea of sequentially adding weak learners is to use the next learner to focus in the error of the

previous one. Therefore, we use them several times each one refocused on the examples that the previous learner found difficult.

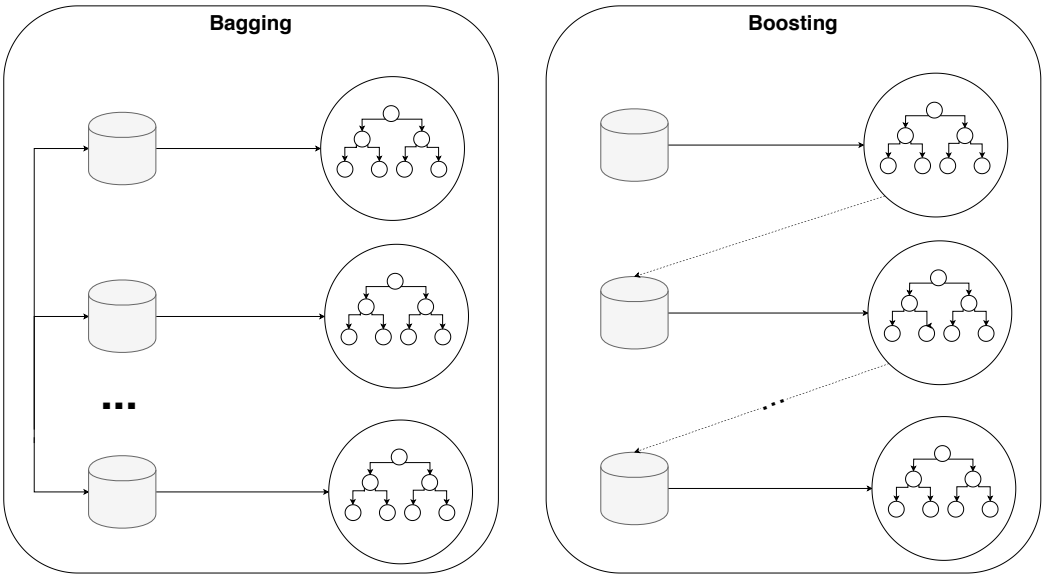


Figure 8 – Comparison between Bagging and Boosting sampling strategies.

The logic behind GBM is simple. First, a model is fitted into the training data, and the residual error is computed by a loss function (usually MSE). Then, the residual error is used to fit the next model, and so on, until the residuals are nearly 0 or some stop criteria is reached. Regularization terms and early stopping are techniques used in GBM to avoid overfitting. In the end, the prediction of each classifier is aggregated.

To improve the GBM performance, XGBoost adds a metric to evaluate the split in each tree, saving memory and computational time. Also, GBM does not support parallelization since the trees are trained sequentially, however with the XGBoost implementation, the construction of a tree is possible to use multiple cores.

3.3 Meta-Learning

The idea of algorithm recommendation problems was introduced by Rice [58], based on the selection of one algorithm from a set of options. Given a set of algorithms $\mathcal{A}$; a set of datasets $\mathcal{P}$ composed of instances from a distribution $\mathcal{D}$ and a performance measure $\mathcal{M} : \mathcal{P} \times \mathcal{A} \rightarrow \mathbb{R}$; the algorithm recommendation problem is to find a function to map $m : \mathcal{P} \rightarrow \mathcal{A}$ that improves the expected performance measure for the problems described in $\mathcal{D}$. Therefore, there are some alternatives to find this mapping function, one of them is through the Meta-Learning (MtL) [59].

The MtL main concept is to use knowledge acquired from previous similar problems to recommend the most suitable algorithm, for a new dataset. Therefore, MtL is the use of

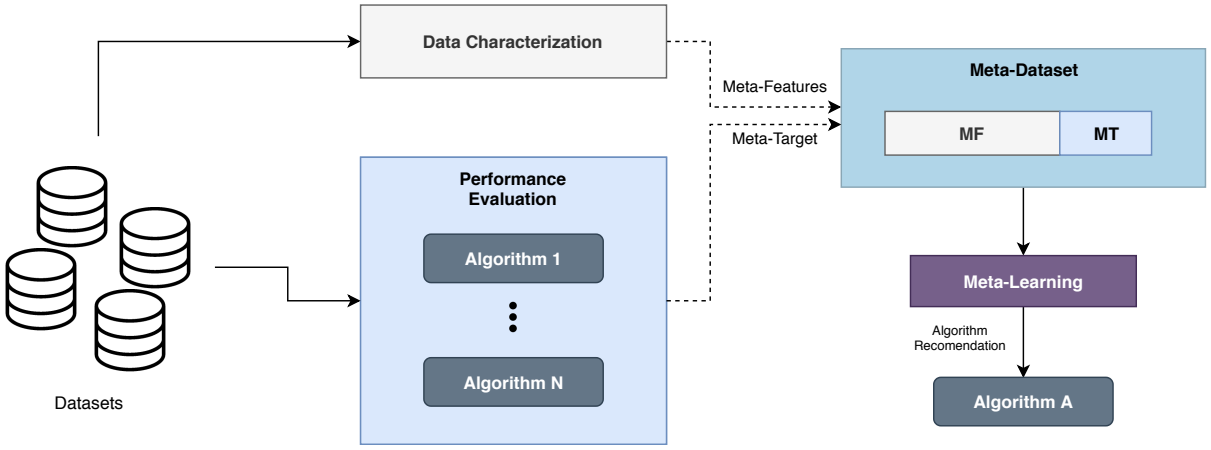


Figure 9 – Overview of the main steps of Meta-Learning. Adapted from Brazdil *et al.* [59]

Machine Learning to map the characteristics of problems (datasets) to the performance of algorithms. Figure 9 presents an overview of the main steps of the Meta-Learning process.

Firstly, the datasets and the algorithms are selected. Each selected dataset is going to be an example in the meta-data. In this work, each dataset would be represented by an image, which will be better explained in Chapter 4. To characterize the datasets, a set of descriptors, named as "meta features", are extracted. At the same time, the algorithms are applied in the datasets, and a performance metric (Accuracy, F-Score, RMSE, *etc*) is used to define which algorithm is the best and selected to be the "meta-target".

Furthermore, as stated by Bradzil [59], it is important to learn what makes an algorithm successful or unsuccessful when performed in a dataset. Also some constraints must be respected to guarantee a learning task in meta-level:

- for each dataset, at least one algorithm should outperform the adopted baseline;
- given a set of algorithms A, their performance at the base level cannot be significantly improved by the addition of a new algorithm;
- when comparing algorithms at the base level, each algorithm should outperform the others at least once;
- an algorithm should not outperform another algorithm in all the datasets.

When these constraints are followed, a valid meta-dataset is generated, and a meta-model is induced through MtL. The meta-model represents the function to map the meta-features, which describes the dataset, to the predictive performance obtained by the algorithm. Consequently, the meta-model may be used to predict the best algorithm for a new unseen problem.

4 MATERIAL AND METHODS

The experiments carried out in this work assess the use of MtL to recommend image segmentation algorithms. Figure 10 provides an overview of the modelling task, using previously extracted knowledge from similar tasks, while Figure 11 presents the recommending tasks. A detailed description of each of their components is presented in the next subsections.

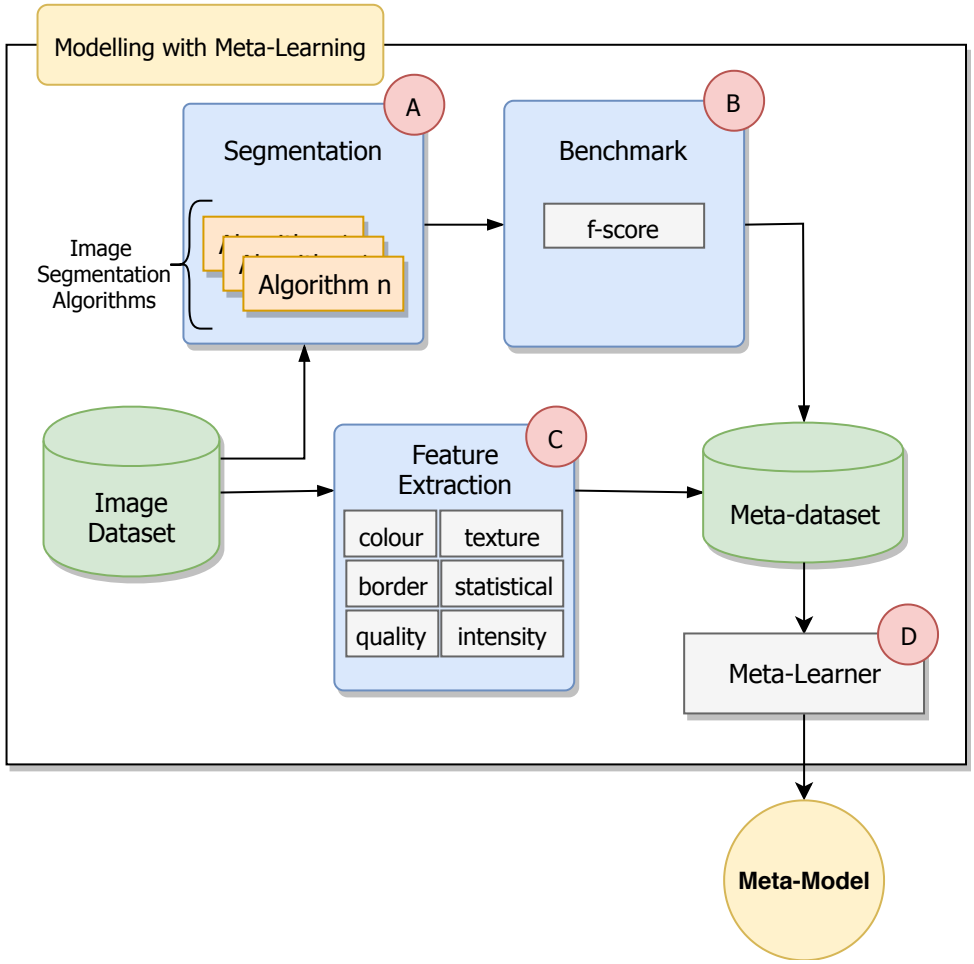


Figure 10 – General modelling overview of the framework.

In the modelling task, a collection of images are segmented by a set of image segmentation algorithms (A) and described by *meta-features* (C). Subsequently, each segmented image is evaluated using the benchmark method proposed by *Martin, Fowlkes and Malik* (2004) [34], which computes the F-score evaluation measure using a ground-truth boundary map (B). The benchmark/evaluation step determines the *meta-targets*, the classes to be predicted by the MtL recommender system. This process builds a meta-dataset.

Finally, ML algorithms (*meta-learners*) are applied to a training subset of the

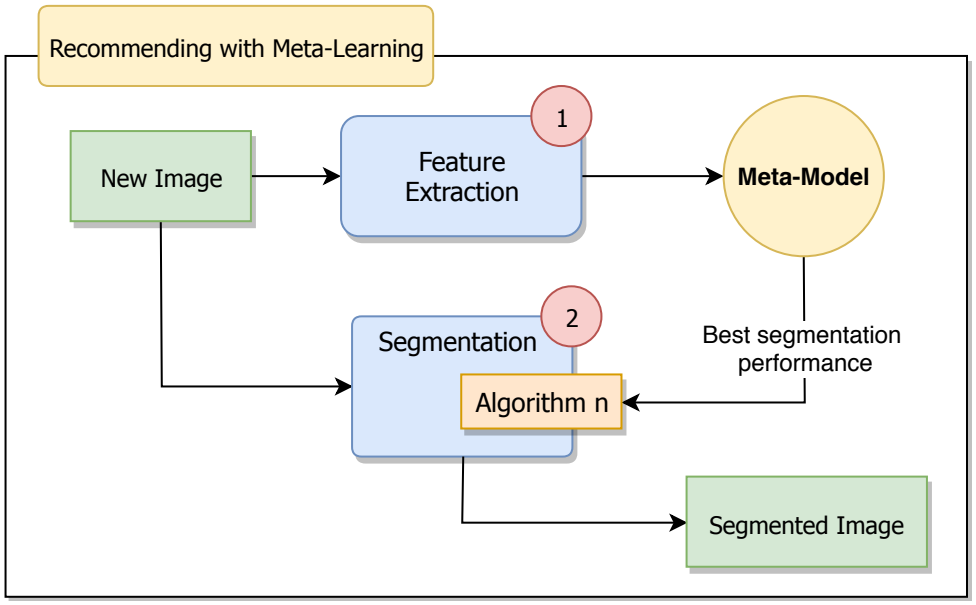


Figure 11 – Meta-learning system to recommend image segmentation algorithms.

meta-dataset to induce a *meta-model*. The meta-model maps the relations between the meta-features and the meta-target. Thus, as illustrated in Figure 11, the meta-model can be applied to a new image, represented by the values of its meta-features, and return the most suitable image segmentation algorithm for that image.

4.1 Meta-dataset

In the experiments, the Berkeley Segmentation Dataset and Benchmark (BSDB500) [60] was used. The BSDB500 is widely known and used for evaluating segmentation algorithms [35, 61, 32, 34]. This dataset is composed of 500 real images divided into three disjoint groups: training (200 images), test (200 images) and validation (100 images). It is important to mention that the train/test/validation approach was made by the authors of the benchmarking dataset and used by the authors of the segmentation algorithms to calibrate them. In the experiments, the validation and test set were combined into a set of 300 images for meta-learning purposes. Since the training set was used to build the ML-based segmentation algorithms, they were not used in the next experiments.

To improve the generalization of the induced meta-models, we created additional learning examples by means of image augmentation. For each original BSDB500 image, we created 7 different distortions by rotating and flipping the image. We made rotations with 90, 180 and 270 degrees around the origin, and flipped the original and rotated images horizontally. At the end, our meta-dataset was composed by 2400 ($300 + 300 \times 7$) images (meta-examples). It is possible to see this process applied to an image of the BSDB500 in Fig. 12.

Eight different segmentation algorithms were applied to each image from BSDB500

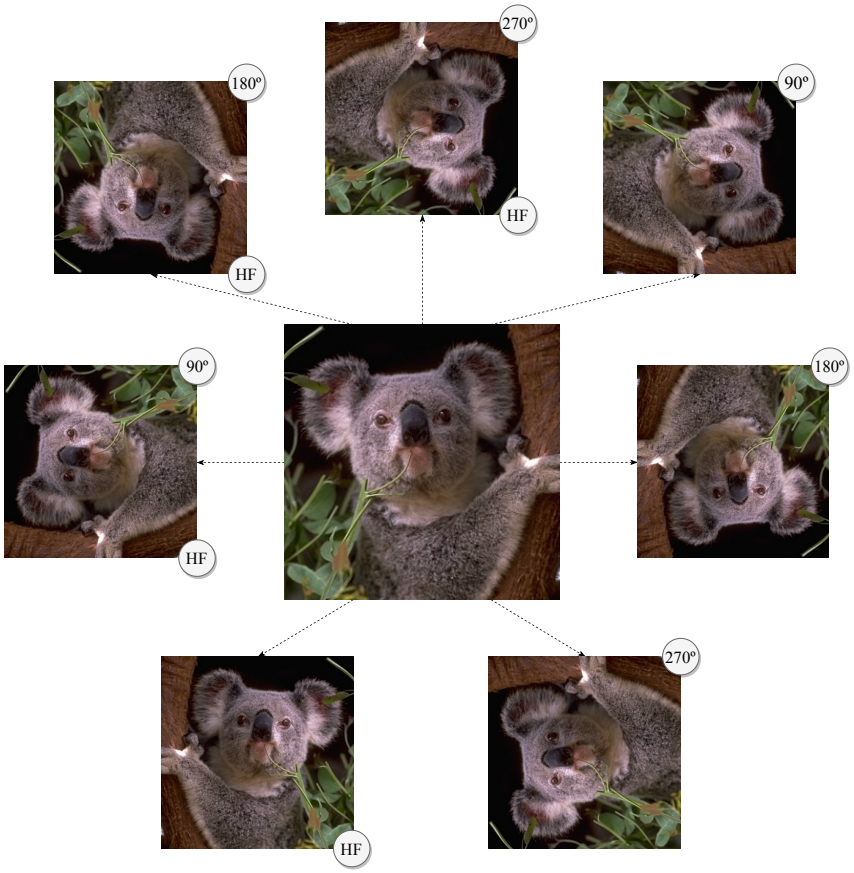


Figure 12 – Process of image augmentation by rotating the original image 3 times and flipping the original and rotated images horizontally.

dataset. The performance obtained by these algorithms was used to define the target feature for each meta-instance in the meta-dataset. Thus, the value of the target of each meta-instance is one of the eight possible labels (segmentation algorithms).

4.2 Meta-features

The characterization of each image by the meta-features produced one meta-instance in the meta-dataset. A meta-feature is a function that extracts a relevant characteristic from an image. The description of each image by a set of meta-features produces a vector of numerical values. In this work, we extracted 94 meta-features from each image. These meta-features can be organized into five groups:

- Colour-based [62]: simple statistical measures from colour channels;
- Border-based: statistical measures obtained after applying border-detector filters;
- Histograms [63]: statistics from histograms of colour and intensity;
- Texture [64]: values from the texture of an image using Fast Fourier Transform (FFT) and Local Binary Patterns (LBP) methods; and

- Image Quality [65]: quality assessment metrics.

Colour-based features provides useful information about the image, being one of the most visually perceived among the feature set. Besides, color characteristics were used as features in supervised classification problems [62]. In this work, 36 colour-based features were used. Two statistical moments (mean and standard deviation) from different channels in three different color spaces: **RGB** (Red, Green, Blue), **HSV** (Hue, Saturation, Value) and **Intensity**. Alongside, the statistical moments, the Spearman correlation value of every channel with all the others was extracted. Also, the entropy of the Intensity channel was used.

Since the objective of the meta-features in our approach is to describe images, in order to select the most suitable segmentation algorithm, and the algorithms are based on edge detection, we introduced Border related features. Canny and Sobel [1] border detectors are applied in the image. Then, the Hu Moments [66] are extracted from Sobel and Canny output image. The Hu Moments are used to shape matching, i.e., it is possible to objectively evaluate similarities between different images by comparing their Moments. The last feature of this group is the number of white pixels in the Sobel Image, which was proposed to measure "how much" of the image is considered boundary.

A histogram is an representation of the distribution of numerical data. In a histogram, the abscissa axis represents a numerical value in the data, and the ordinate axis the frequency of that value. In digital image processing, the histogram is a useful tool for characterization [1]. In this sense, based on the histogram of each channel (RGB, HSV and Intenisty) we extracted the second (standard deviation), third (skewness) and fourth (kurtosis) statistical moments [62].

Image texture features provide information about the spatial arrangement of intensities in an image. It was applied in different image-classification applications [67]. In our work, we selected three commonly used texture descriptors: Local Binary Patterns (LBP) [68], Gray Level Co-Occurrence Matrix (GLCM) [67] and Fast Fourier Transform (FFT) frequency domains features. In the FFT frequency domain, 4 statistical metrics were extracted: Energy, Entropy, Intertia and Homogeneity. Regarding the GLCM the same metrics were extracted, and also, the correlation.

Image Quality Assessment (IQA) techniques are used for image quality automatic evaluation according to how a human would evaluate. Thus, two IQA techniques were selected to compose our meta-features: Statistical Naturalness Measure (SNM) [65] and Measure of Enhancement (EME) [69].

A summarized list of the meta-features used in our experiments can be seen in Table 2. The complete list with the description of each meta-feature is in Table 4 in the appendix.

Table 2 – Category, acronym and description of meta-features used in the experiments.

Category	**Acronym**	**Description**
Colour-based	cor_*	Spearman correlation value between * channels pairwise (RGB, HSV and Intensity)
	mean_*	Mean of the * channel (RGB, HSV)
	std_*	Standard deviation of the * channel (RGB, HSV and Intensity)
	entropy_I	Entropy of the Intensity Channel
Image Quality	SNM	Statistical Naturalness Measure
	EME	Measure of Enhancement
Border	nump_sobel	Number of white pixels in a Sobel Image
	hu_sobel[1-7]	Hu Moments of Sobel Image
	hu_canny[1-7]	Hu Moments of Canny
Histogram	std_hist_*	Standard deviation of the histogram of * channel (RGB, HSV and Intensity)
	kurt_hist_*	Kurtosis of the histogram of * channel (RGB, HSV and Intensity)
	skew_hist_*	Skewness of the histogram of * channel (RGB, HSV and Intensity)
Texture	lbp [0-9]	LBP Vector
	com_entropy	Entropy of Co-occurence Matrix
	com_inertia	Inertia of Co-occurence Matrix
	com_energy	Energy of Co-occurence Matrix
	com_correlation	Correlation of Co-occurence Matrix
	com_homogeniety	Homogeneity of Co-occurence Matrix
	FFT_energy	Energy of FFT
	FFT_entropy	Entropy of FFT
	FFT_intertia	Inertia of FFT
	FFT_homogeneity	Homogeneity of FFT

4.3 Meta-targets

The value of the meta-target defines which image segmentation algorithm is the most suitable for a given image. The eight segmentation algorithms applied to each image can produce segmentation with different qualities and demand different processing times.

As the main goal of this work is to recommend the best algorithm for each new image, we avoided the comparison of algorithms with very different computational complexities. Thus, we divided the segmentation algorithms into two groups:

- ML-based algorithms: includes the algorithms with high computationally complexity, in this study: Sparse Code Gradients (SPCs) [31], Global Probability of Boundary (gPb) [33] and Global Probability of Boundary Ultrametric Contour Maps (gPB-ucm) [32]. These algorithms are based on ML;
- Gradient-based algorithms: algorithms with low complexity, based on Colour, Texture and Brightness Gradients: Colour Gradient (CG), Texture Gradient (TG), Brightness Gradient (BG), Colour/Texture Gradient (CGTG), Brightness/Texture

Gradient (BGTG) [34].

In Figure 13, it can be seen one example of each segmentation algorithm applied in the same image.

In the evaluation of the segmentation algorithms, we used the methodology proposed by *Martin, Fowlkes and Malik* (2004) [34]. For each image in BSDB500, there are ground truth boundaries provided by human segmented images. We assume any boundary marked by a human to be valid. Then, our goal is to determine the similarity between the boundary map which is the output of the segmentation algorithm and the ground truth boundaries.

The straightforward approach to compare the boundary map and the ground truth is to binarize the boundary map by choosing some threshold. However, to select the optimal threshold point depends on the domain, and the evaluation should be useful across different domains. Also, thresholding boundaries is likely to be a bad idea, since it destroys much information. Bearing this in mind, the evaluation is based on a non-fixed-thresholded boundary map. However, to do the comparison, we need to threshold the boundary map, but it is done at 30 levels, using a variable value of threshold. For each level a value l is defined as $l = 1/n$ where $n \in (1, 30)$. Then, we binarize the boundary map with every different value of l.

Next, for each level, it is computed two metrics: (i) precision and (ii) recall. Precision is the probability that the pixel in the probability map is a true boundary pixel. Recall is the probability that a true boundary pixel is detected. Then, we compute the F-score that is the harmonic mean of the precision and recall. It is defined for all levels, and the maximum F-score value is reported as the F-score of the segmentation algorithm.

The algorithm with the highest F-score in an image becomes the image label (meta-target) value. Also, it is important to endorse that the augmented images have their own F-scores, not necessarily the same of the original image. In Figure 14 and 15, it is possible to see the distribution of the meta-targets for each image and its associated augmented. Figures 14 and 15 present the distribution of meta-targets by augmented image, indicating that changing the orientation of the image impacts in the segmentation performance.

Table 3 presents the class distribution for each group of algorithms. It is important to mention that both scenarios are imbalanced since classes differ substantially in the number of examples because one of the algorithms is the best segmentation algorithm for a large number of images. Considering the superior number of algorithms (meta-targets) from both meta-datasets, and to deal with the high imbalanced meta-dataset, we applied under-sampling for both meta-datasets. Since the minority class in the Gradient-based (BG) had 179 wins, 170 images were randomly selected for each class. The same approach was adopted for the ML-based group. Thus, the Gradient-based meta-dasaset

was composed with 850 examples (170 images × 5 algorithms.) and the ML-based with 510 examples (170 images × 3 algorithms). It is worth to mention that the undersampling process does not change the nature of the problem. When we look to related studies, we can see that the problem of segmentation algorithm selection is not intrinsically imbalanced. Since we have a potentially infinite number of images, and an uncountable number of segmentation algorithms proposed for different scenarios, and by the *'No free lunch'* [**??**] theorem, no algorithm would be the best in most of the scenarios, leading to a more balanced problem.

Table 3 – Meta-datasets of each group, the meta-targets and the number of times (and %) they were the most suitable in the group.

Meta-Dataset	**Meta-Target**	**Number of Wins**	**Number of Meta-examples**	**Percentage of Wins**
Gradient-based	CGTG	1495	170	62.29
	BGTG	331	170	13.79
	TG	202	170	8.42
	CG	193	170	8.04
	BG	179	170	7.46
ML-based	gPb-ucm	1348	170	56.17
	SPC	879	170	36.63
	gPb	173	170	7.21

4.4 Meta-learners

Three ML algorithms were used as meta-learners: RF [53], SVM [55] and XGBoost [56]. They induce models following different learning biases and have been used with success in multiple predictive tasks. They were implemented using the `R` language[1] and the `mlr` package[2] [70] along with their default hyperparameter values.

4.5 Performance Evaluation

The predictive performance of the meta-learners was evaluated using the Leave-One-Out Cross Validation (LOO-CV) strategy. Two different baselines were also used in the experimental comparisons: a model that always recommends a unique class for the whole dataset (`OneR`) and a model that provides random recommendations (`Random`). Additionally, we used an upperbound as the ground-truth (`Truth`), related to the segmentation algorithm that obtained the best segmentation performance according to [34].

Seven evaluation metrics were used to assess the predictive performance of the induced models: Accuracy, Precision, Recall, F-score (f1), Sensitivity, Specificity and Balanced per class accuracy. Considering that we are dealing with a multi-class problem, since

[1] https://www.r-project.org
[2] https://mlr-org.github.io/mlr-tutorial/release/html/index.html

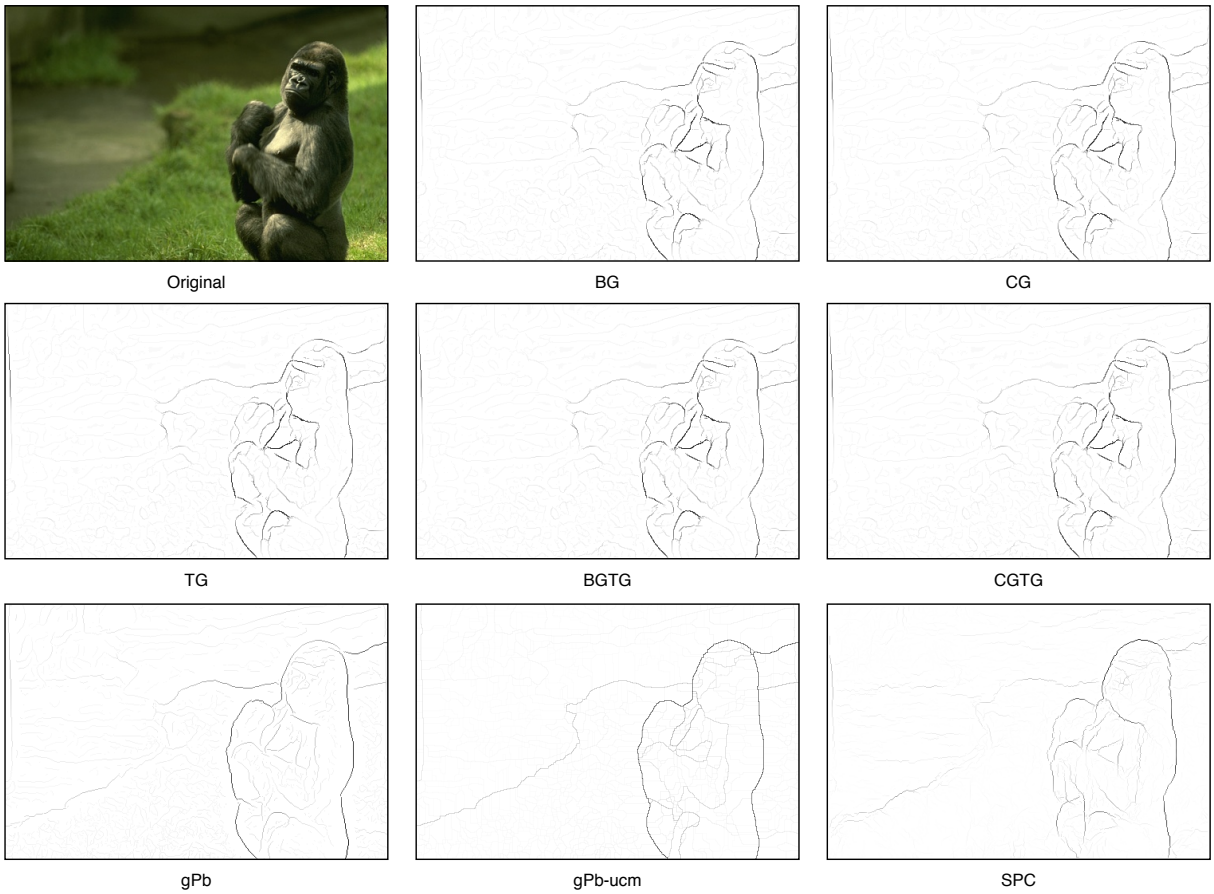


Figure 13 – Example of segmentation of each segmentation algorithm.

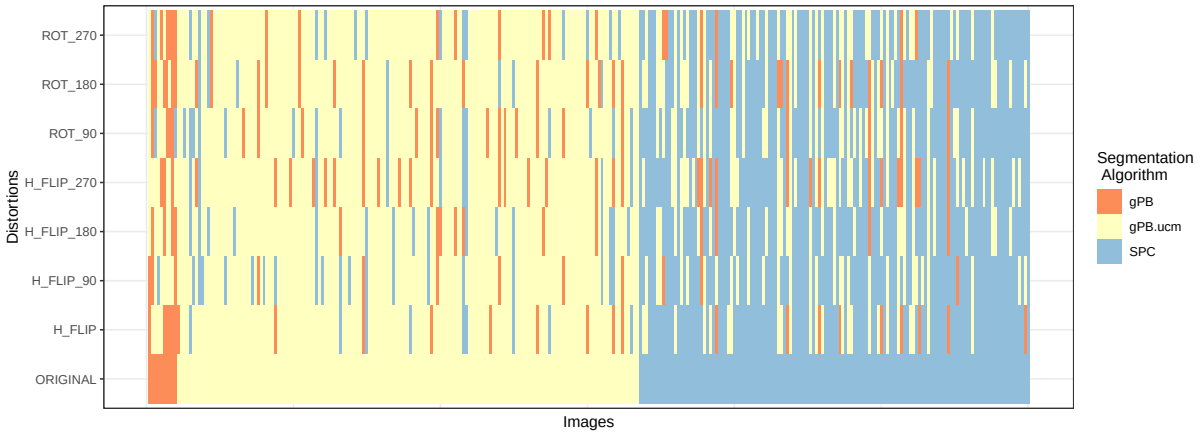


Figure 14 – Machine Learning-based group meta-targets associated with each image and its augmented ones.

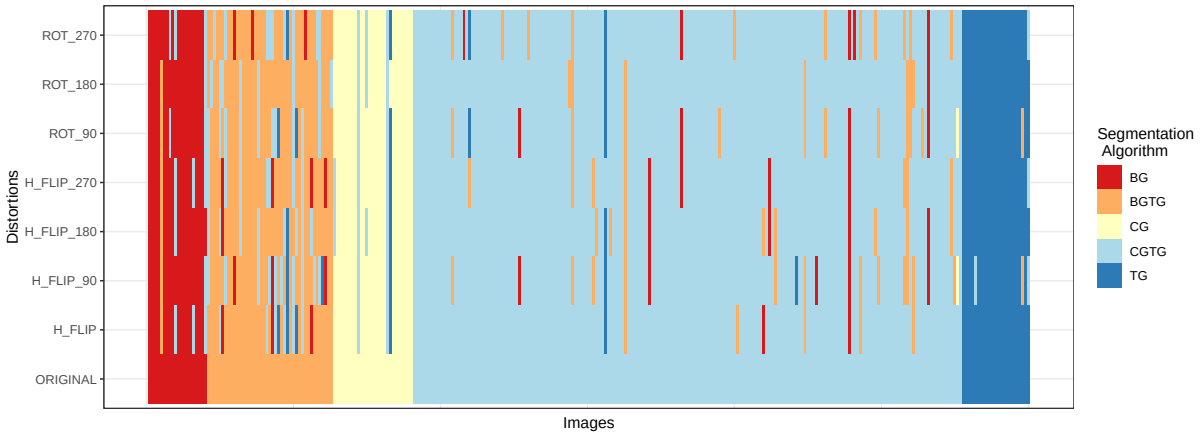


Figure 15 – Gradient-based group meta-targets associated with each image and its augmented ones.

we have more than 2 classes to predict, only accuracy is obtained using the predictions of every class. All the other metrics are obtained by class, and then the average value is calculated.

5 RESULTS

The radar chart in Figure 16 and Figure 17 presents the predictive performance obtained by the meta-learners in the test subset of the two meta-datasets. In these figures, each line represents a meta-model and each vertex accounts for a different performance measure. The larger the area in the radar chart, the better the meta-model.

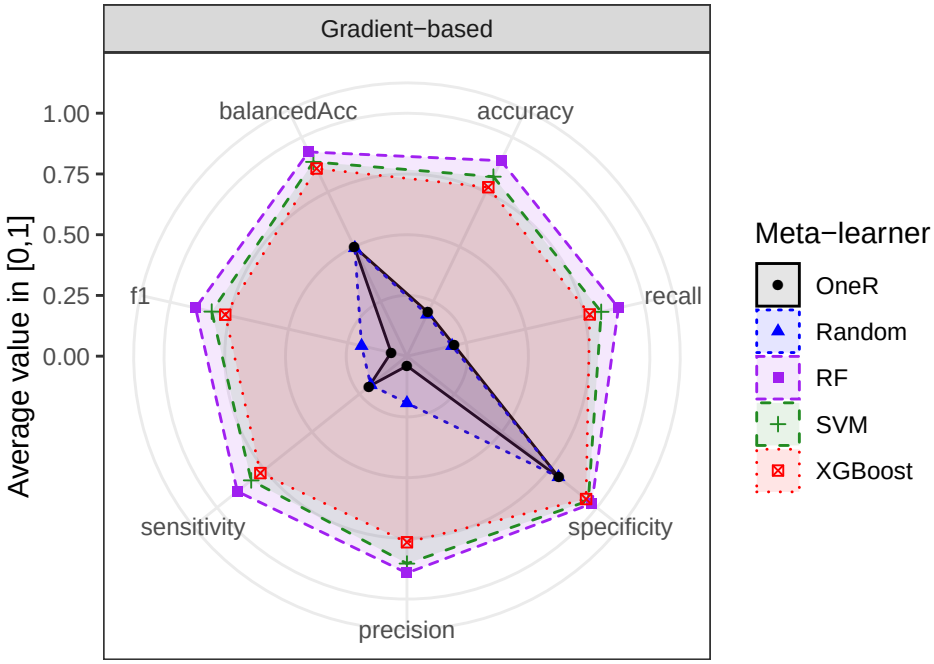


Figure 16 – Performance of the meta-models for Gradient-based algorithms

Regarding the Gradient-based meta-datasets, the radar charts shows that the meta-models outperformed the baselines. The best segmentation algorithm was not recommended only in 51 out of 850 meta-instances (6%). All meta-models overcame the `OneR` and `Random` baselines for both datasets. It is also possible to see in the results that RF obtained the best results with 89.2% of accuracy, precision, and recall and 0.89 of F-score.

The predictive performances for the ML-based meta-dataset show that in the ML-based group, considering all meta-models, only in 109 out of 510 (21.3%) meta-instances the best image segmentation algorithm was not recommended. This reduction, when compared with the Gradient-based meta-datasets, occurred because the predictive performance of the baselines are closer to the predictive performance of the ML-based segmentation algorithms. The results also show that the meta-model induced by RF obtained

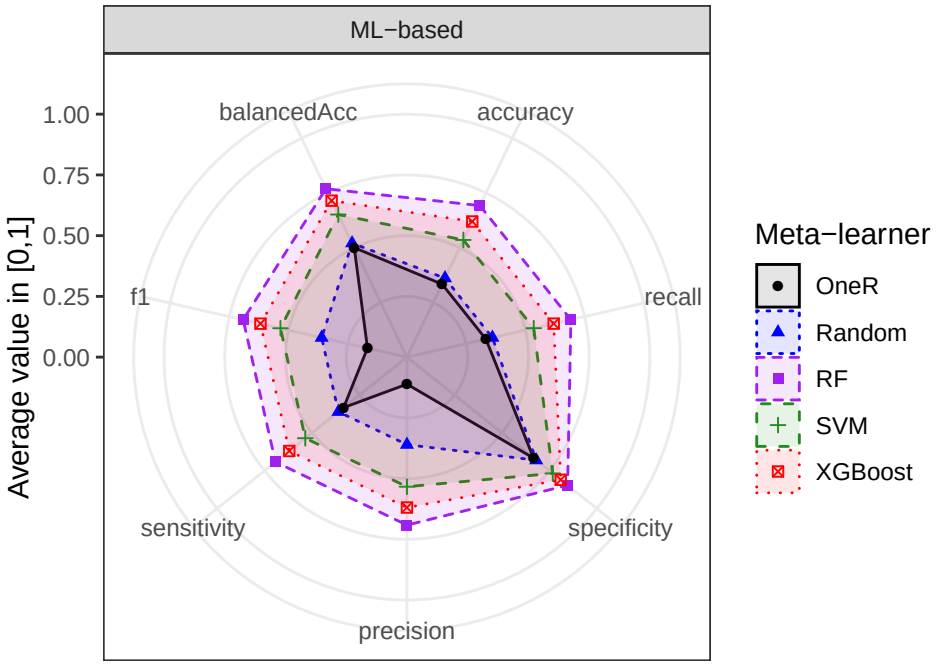


Figure 17 – Performance of the meta-models for ML-based algorithms

the best predictive performance, with 69% of accuracy, precision, and recall and 0.69 of F-score.

Observing the number of misclassified images by distortion of the ML-based set, i.e., given an original image, how many of their distortions were misclassified, for 11 images the meta-models misclassified all of its distortions, however, for just one image there were more than 2 distortions in the sub-sampled dataset. Then, for only one image (`16077.jpg`) the meta-models misclassified all the 4 distortions. Similarly, in the experiments with the Gradient-based set, just one sample (`170057.jpg`) was misclassified by the meta-models regardless of the type of image augmentation performed.

The superiority of the MtL recommending system in relation to the baselines was confirmed by statistical tests. We used the Friedman test, with a significance level of $\alpha = 0.05$. The null hypothesis is that the recommendation by the meta-models and by the baselines are similar. Anytime the null hypothesis is rejected, the Nemenyi post hoc test can be applied, stating that the performance of the two approaches are significantly different if their corresponding average ranks differ by at least a Critical Difference (CD) value. When multiple algorithms are compared in this way, a graphic representation can be used to represent the results with the CD diagram, as proposed in [71].

Considering the Gradient-based set, the meta-models (RF, SVM, XGBoost) were compared with `Truth` (expected algorithm suggestion), the use of the single segmentation

algorithms as `OneR` (CG, TG, BG, CGTG and BGTG), and the random selection of segmentation algorithm for each image (Random), using their F-Score values as performance metric. This analysis is shown in Fig. 18, using the results from the Nemenyi test.

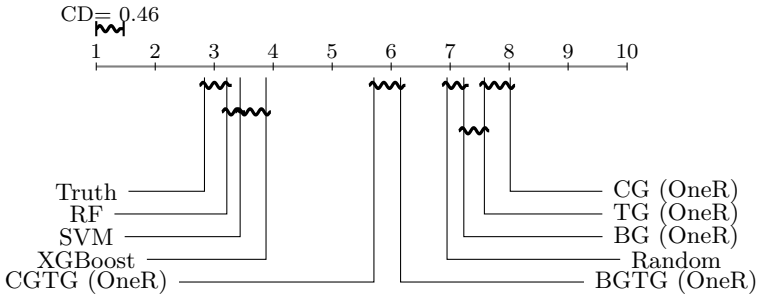


Figure 18 – Comparison of the F-Score values obtained by meta-models when recommending Gradient-based segmentation algorithms according to the Nemenyi test. Groups of meta-learners that are not significantly different (α= 0.05 and CD = 0.46) are connected.

As shown in Fig. 18, the RF meta-model is connected to the `Truth` baseline at the top of the ranking. Thus, this meta-model recommends the best segmentation algorithm, which is statistically similar to `Truth`, it implies that even with some misclassified examples the overall segmentation performance of the predictions are equivalent with to test every segmentation algorithm and use always the best. RF, SVM, XGBoost are connected, supporting the benefit of using MtL in comparison to select a specific algorithm to the whole dataset. When applying a single segmentation algorithm (`OneR`), CGTG and BGTG were superior to the others. In particular, the CG, TG and BG algorithms were similar to a random choice of algorithms, presenting the worst F-score performance.

Evaluating the ML-based recommending using `Truth`, meta-models (RF, SVM, and XGBoost), `OneR` (gPb-ucm, SPC and gPb) and `Random`, Figure 19 shows that no solution was similar to the expected `Truth` value. However, the meta-model recommending results were superior to `Random` or `OneR` with RF providing the best recommendations, with similar results for SVM and XGBoost.

Finally, the segmentation performance obtained by the segmentation algorithms

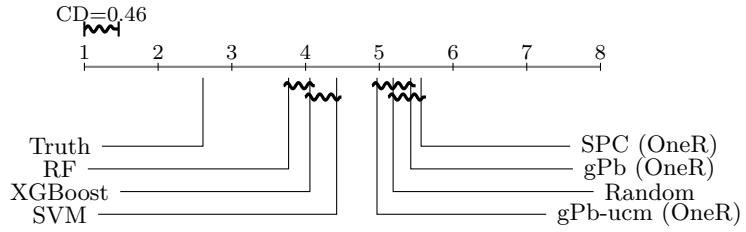


Figure 19 – Comparison of the F-score values obtained by meta-models when recommending ML-based segmentation algorithms according to the Nemenyi test. Groups that are not significantly different (α= 0.05 and CD = 0.46) are connected.

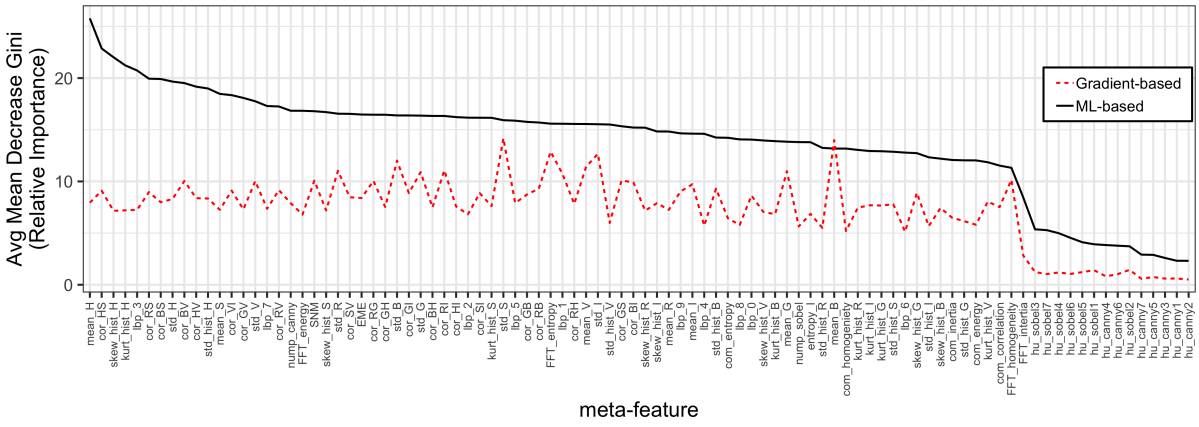


Figure 20 – Average relative importance of the meta-features obtained from RF importance. The names of the meta-features in the x-axis follow the acronyms presented in Tables 2 sorted by the most important features for the ML-based set.

recommended by meta-models was superior to always using the same segmentation algorithm or a segmentation algorithm randomly chosen. Regarding the predictive performance of the meta-models, RF induced the best meta-models in both sets of segmentation algorithms. Moreover, when recommending Gradient-based algorithms, RF was similar to the ground-truth solution.

5.1 Relative feature importance

It is also possible to assess the importance of each image feature for the induction of the meta-models by using the RF Feature Importance metric. This inner RF's metric is calculated by permuting the values of a feature in the Out-of-Bag (OOB) samples and recalculating the OOB Error. In this way, if replacing the values of a feature by random values results in error increase, this feature was considered important. Otherwise, if the error decreases, the resulting importance is negative, and the feature is considered not important [53].

We used the RF Feature Importance to investigate the contribution of each feature in selecting segmentation algorithms. Fig. 20 shows the feature importance for both datasets. In the Gradient-based meta-dataset, the Texture and Colour meta-features were considered the most important, especially FFT Entropy (12.88), Mean (13.99) and Standard Deviation of Blue Channel (12.01), and Standard Deviation of Saturation Channel (14.184). Once the Gradient-based algorithms are grounded on information from Texture and Colour attributes of image regions to segment images, their selection makes sense.

ML-based segmentation algorithms were suggested by meta-models with high meta-feature importance (>20) from *mean_h* (25.77), *cor_hs* (22.86), *skew_hist_H*(22.01), *kurt_hist_H*(21.22) and *lbp_*3(20.73). Some of them are extracted from H colour channel

and histogram. However, the high importance of meta-features from LBP Vector (Texture category) suggests that a more precise segmentation algorithm, supported by ML-based, requires a plural of image descriptors. Another example was SNM from Image Quality (twentieth most important with 16.79 of importance), which was more important than other colours, histogram and intensity features.

In both meta-datasets, the border meta-features ($hu\_sobel\{1..7\}$ and $hu\_canny\{1..7\}$) had low importance at the point of being removed from the meta-features without compromising the results.

6 CONCLUSION

This work presented an approach for solving the problem of selecting the most suitable segmentation algorithm for a given image. The problem was approached by using the concept of Meta-Learning (MtL) that has been applied in the Machine Learning field for algorithm selection and hyperparameter tuning.

Therefore, it was presented a framework to recommend segmentation algorithms using MtL for automatic image segmentation. A set of 94 meta-features was used to describe the input image and provide the information for the induced meta-model to predict which segmentation algorithm would achieve the best performance.

The experiments were carried out using the Berkeley Segmentation Dataset and Benchmark (BSDB500). This dataset is composed of 300 images from general domains and used for segmentation algorithm benchmarking. For each image, two segmentation algorithms were suggested, from a set of 8 algorithms, one from the ML-based and the other from the Gradient-based group. The ML-based group is composed with 3 algorithms with high computationally costly solutions and with high segmentation performance. On the other hand, the Gradient-based group is composed of less complex and faster techniques based on colour, texture and brightness. All the algorithms were evaluated in the image using an objective evaluation, not depending on an expert opinion.

Three Machine Learning (ML) algorithms were compared for meta-model induction. For both groups of segmentation algorithms, the meta-models were superior to both baselines: the `OneR` and `Random` models. In the Gradient-based group, the best predictive performance was achieved by the RF meta-model, followed by the SVM and the XGBoost. Considering the ML-group, the RF also presented the best results.

The superiority of the recommendation system based on MtL regarding the baselines was assessed using the Nemenyi post-hoc test, showing that is better to use the recommended algorithm than choosing a random algorithm or choosing the same one for every task.

This work also investigated the importance of the meta-features given by the RF algorithm. The results of the performed analysis highlight colour and histogram meta-features as the most important for Gradient-based segmentation algorithms. The analysis also showed that the recommendation of the ML-based segmentation algorithms considered important meta-features from different categories.

Thus, the proposed approach showed that it is possible to recommend the best segmentation algorithm using Meta-Learning (MtL) for different image domains with high segmentation performance.

As future work, we intend to use features extracted using Convolutional Neural Networks (CNN), instead of handcrafted features. We believe that these meta-features will increase the predictive performance. We also plan to increase the number of segmentation algorithms used. CNNs could also be used to evaluate if a segmentation was adequate or not. Finally, we plan to increase the number of segmentation algorithms in future experiments.

BIBLIOGRAPHY

[1] GONZALEZ, R. C.; WOODS, R. E. et al. Digital image processing [m]. *Publishing house of electronics industry*, v. 141, n. 7, 2002.

[2] SCHAEFER, G. et al. Colour and contrast enhancement for improved skin lesion segmentation. *Computerized Medical Imaging and Graphics*, Elsevier, v. 35, n. 2, p. 99–104, 2011.

[3] AVENDI, M.; KHERADVAR, A.; JAFARKHANI, H. A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri. *Medical image analysis*, Elsevier, v. 30, p. 108–119, 2016.

[4] BARBIN, D. F. et al. Digital image analyses as an alternative tool for chicken quality assessment. *Biosystems Engineering*, Elsevier, v. 144, p. 85–93, 2016.

[5] CHEN, K.; QIN, C. Segmentation of beef marbling based on vision threshold. *computers and electronics in agriculture*, Elsevier, v. 62, n. 2, p. 223–230, 2008.

[6] DUTTA, M. K. et al. Image processing based method to assess fish quality and freshness. *Journal of Food Engineering*, Elsevier, v. 177, p. 50–58, 2016.

[7] DOLLAR, P. et al. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 34, n. 4, p. 743–761, 2011.

[8] FERNANDEZ, M. A.; LOPES, R. M.; HIRATA, N. S. Image segmentation assessment from the perspective of a higher level task. In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*. [S.l.], 2015. p. 111–118.

[9] ZAITOUN, N. M.; AQEL, M. J. Survey on image segmentation techniques. *Procedia Computer Science*, Elsevier, v. 65, p. 797–806, 2015.

[10] VALA, M. H. J.; BAXI, A. A review on otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, v. 2, n. 2, p. pp–387, 2013.

[11] CAMPOS, G. F. C.; BARBON, S.; MANTOVANI, R. G. A meta-learning approach for recommendation of image segmentation algorithms. In: *29th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2016, Sao Paulo, Brazil, October 4-7, 2016*. IEEE Computer Society, 2016. p. 370–377. ISBN 978-1-5090-3568-7. Disponível em: <https://doi.org/10.1109/SIBGRAPI.2016.058>.

[12] IONESCU, R. T. et al. How hard can it be? estimating the difficulty of visual search in an image. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 2157–2166.

[13] ZHANG, K. et al. A level set approach to image segmentation with intensity inhomogeneity. *IEEE transactions on cybernetics*, IEEE, v. 46, n. 2, p. 546–557, 2016.

[14] KRONMAN, A.; JOSKOWICZ, L. Image segmentation errors correction by mesh segmentation and deformation. In: SPRINGER. *International Conference on Medical Image Computing and Computer-Assisted Intervention*. [S.l.], 2013. p. 206–213.

[15] WOLPERT, D. H. The lack of a priori distinctions between learning algorithms. *Neural computation*, MIT Press, v. 8, n. 7, p. 1341–1390, 1996.

[16] YONG, X. et al. Learning-based algorithm selection for image segmentation. *Pattern Recognition Letters*, v. 26, n. 8, p. 1059–1068, 2005. ISSN 01678655.

[17] TAKEMOTO, S.; YOKOTA, H. Algorithm selection for intracellular image segmentation based on region similarity. In: IEEE. *2009 Ninth International Conference on Intelligent Systems Design and Applications*. [S.l.], 2009. p. 1413–1418.

[18] BRAZDIL, P. et al. *Metalearning: Applications to Data Mining*. 2. ed. [S.l.]: Springer Verlag, 2009.

[19] LEMKE, C.; BUDKA, M.; GABRYS, B. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, v. 44, n. 1, p. 117–130, Jun 2015. ISSN 1573-7462.

[20] ROSSI, A. L. D. et al. Metastream: A meta-learning based method for periodic algorithm selection in time-changing data. *Neurocomputing*, Elsevier, v. 127, p. 52–64, 2014.

[21] BRAZDIL, P. B.; SOARES, C.; COSTA, J. P. D. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, Springer, v. 50, n. 3, p. 251–277, 2003.

[22] REIF, M.; SHAFAIT, F.; DENGEL, A. Meta-learning for evolutionary parameter optimization of classifiers. *Machine Learning*, Springer US, v. 87, p. 357–380, 2012.

[23] FINN, C.; ABBEEL, P.; LEVINE, S. Model-agnostic meta-learning for fast adaptation of deep networks. In: JMLR. ORG. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. [S.l.], 2017. p. 1126–1135.

[24] SANTORO, A. et al. Meta-learning with memory-augmented neural networks. In: *International conference on machine learning*. [S.l.: s.n.], 2016. p. 1842–1850.

[25] ATTIG, A.; PERNER, P. A study on the case image description for learning the model of the watershed segmentation. *Transactions on Case-Based Reasoning*, ibai Publishing, v. 2, n. 1, p. 41–53, 2009.

[26] MARAKEBY, H. A.; ZAKI, M.; SAMIR, I. S. A generalized object detection system using automatic feature selection. In: IEEE. *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*. [S.l.], 2010. p. 839–844.

[27] YAMAGUCHI, T. et al. Application of particle swarm optimization to similar image search on satellite sensor data. In: IEEE. *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*. [S.l.], 2012. p. 1573–1577.

[28] OTSU, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 9, n. 1, p. 62–66, 1979.

[29] LI, H.; HE, H.; WEN, Y. Dynamic particle swarm optimization and k-means clustering algorithm for image segmentation. *Optik-International Journal for Light and Electron Optics*, Elsevier, v. 126, n. 24, p. 4817–4822, 2015.

[30] WANG, X.-Y.; WANG, T.; BU, J. Color image segmentation using pixel wise support vector machine classification. *Pattern Recognition*, Elsevier, v. 44, n. 4, p. 777–787, 2011.

[31] XIAOFENG, R.; BO, L. Discriminatively trained sparse code gradients for contour detection. *Neural Information Processing Systems*, p. 593–601, 2012. ISSN 10495258.

[32] ARBELAEZ, P. et al. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 33, n. 5, p. 898–916, 2011.

[33] MAIRE, M. et al. Using contours to detect and localize junctions in natural images. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008. ISSN 0891-6640.

[34] MARTIN, D. R.; FOWLKES, C. C.; MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 26, n. 5, p. 530–549, 2004.

[35] ACHANTA, R. et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, Institute of Electrical and Electronics Engineers, Inc., 345 E. 47 th St. NY NY 10017-2394 United States, v. 34, n. 11, p. 2274–2282, 2012.

[36] MITTAL, A.; MOORTHY, A. K.; BOVIK, A. C. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, IEEE, v. 21, n. 12, p. 4695–4708, 2012.

[37] TU, Z.; ZHU, S.-C. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 5, p. 657–673, 2002.

[38] ALI, S.; SMITH-MILES, K. A. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, v. 70, n. 13, p. 173–186, 2006.

[39] CAMPOS, G. F. et al. Supervised approach for indication of contrast enhancement in application of image segmentation. In: *MMEDIA 2016, The Eighth International Conferences on Advances in Multimedia*. [S.l.: s.n.], 2016. p. 12–18.

[40] PEREIRA, L. F. S. et al. Predicting the ripening of papaya fruit with digital imaging and random forests. *Computers and Electronics in Agriculture*, Elsevier, v. 145, p. 76–82, 2018.

[41] BROSNAN, T.; SUN, D.-W. Improving quality inspection of food products by computer vision—-a review. *Journal of food engineering*, Elsevier, v. 61, n. 1, p. 3–16, 2004.

[42] COIFMAN, B. et al. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*, Elsevier, v. 6, n. 4, p. 271–288, 1998.

[43] KUMAR, N. et al. Leafsnap: A computer vision system for automatic plant species identification. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2012. p. 502–516.

[44] UMBAUGH, S. E. *Digital image processing and analysis: human and computer vision applications with CVIPtools*. [S.l.]: CRC press, 2010.

[45] CHAUHAN, A. S.; SILAKARI, S.; DIXIT, M. Image segmentation methods: A survey approach. In: IEEE. *2014 Fourth International Conference on Communication Systems and Network Technologies*. [S.l.], 2014. p. 929–933.

[46] FAIRCHILD, M. D. *Color appearance models*. [S.l.]: John Wiley & Sons, 2013.

[47] FOWLKES, C. C.; MALIK, J. *How much does globalization help segmentation?* [S.l.]: Computer Science Division, University of California, 2004.

[48] HEIJMANS, H. J. *Morphological image operators*. [S.l.]: Academic Press Boston, 1994. v. 4.

[49] AHARON, M.; ELAD, M.; BRUCKSTEIN, A. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, IEEE, v. 54, n. 11, p. 4311–4322, 2006.

[50] PATI, Y. C.; REZAIIFAR, R.; KRISHNAPRASAD, P. S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: IEEE. *Proceedings of 27th Asilomar conference on signals, systems and computers*. [S.l.], 1993. p. 40–44.

[51] PERRONNIN, F.; SÁNCHEZ, J.; MENSINK, T. Improving the fisher kernel for large-scale image classification. In: SPRINGER. *European conference on computer vision*. [S.l.], 2010. p. 143–156.

[52] BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006.

[53] BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001.

[54] BREIMAN, L. *Classification and regression trees*. [S.l.]: Routledge, 2017.

[55] VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013.

[56] CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: ACM. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.], 2016. p. 785–794.

[57] FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001.

[58] RICE, J. R. The algorithm selection problem. In: *Advances in computers*. [S.l.]: Elsevier, 1976. v. 15, p. 65–118.

[59] BRAZDIL, P. et al. *Metalearning: Applications to Data Mining*. 2. ed. [S.l.]: Springer Verlag, 2009.

[60] MARTIN, D. et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. 8th Int'l Conf. Computer Vision*. [S.l.: s.n.], 2001. v. 2, p. 416–423.

[61] SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Science & Business Media, 2010.

[62] LI, D. et al. Pornographic images recognition based on spatial pyramid partition and multi-instance ensemble learning. *Knowledge-Based Systems*, Elsevier, v. 84, p. 214–223, 2015.

[63] BALAJI, G.; SUBASHINI, T.; CHIDAMBARAM, N. Automatic classification of cardiac views in echocardiogram using histogram and statistical features. *Procedia Computer Science*, Elsevier, v. 46, p. 1569–1576, 2015.

[64] HARALICK, R. M.; SHAPIRO, L. G. Image segmentation techniques. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Applications of Artificial Intelligence II*. [S.l.], 1985. v. 548, p. 2–10.

[65] YEGANEH, H.; WANG, Z. Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, IEEE, v. 22, n. 2, p. 657–667, 2013.

[66] HU, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, IEEE, v. 8, n. 2, p. 179–187, 1962.

[67] HARALICK, R. M.; SHANMUGAM, K. et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, Ieee, n. 6, p. 610–621, 1973.

[68] OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 7, p. 971–987, 2002.

[69] AGAIAN, S. S.; PANETTA, K.; GRIGORYAN, A. M. Transform-based image enhancement algorithms with performance measure. *IEEE Transactions on Image Processing*, IEEE, v. 10, n. 3, p. 367–382, 2001.

[70] BISCHL, B. et al. mlr: Machine learning in r. *Journal of Machine Learning Research*, v. 17, n. 170, p. 1–5, 2016.

[71] DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, v. 7, n. Jan, p. 1–30, 2006.

Appendix

APPENDIX A – EXTRA TABLES

Table 4 – Meta-features used in the experiment

id	Category	Name	Description	id	Category	Name	Description
1	Colour-based	cor_RG	Spearman correlation value between Red and Green channel	38	Image Quality	EME	Measure of Enhancement
2	Colour-based	cor_RB	Spearman correlation value between Red and Blue channel	39	Border	nump_sobel	Number of white pixels in a Sobel Image
3	Colour-based	cor_RH	Spearman correlation value between Red and Hue channel	40	Border	nump_canny	Number of white pixels in a Canny Image
4	Colour-based	cor_RS	Spearman correlation value between Red and Saturation channel	41-47	Border	hu_sobel[1-7]	Hu moments of Sobel Image
5	Colour-based	cor_RV	Spearman correlation value between Red and Value channel	48-54	Border	hu_canny1[1-7]	Hu moments of Canny Image
6	Colour-based	cor_RI	Spearman correlation value between Red and Intensity channel	55	Histogram	std_hist_H	Standard deviation of the histogram of Hue channel
7	Colour-based	cor_GB	Spearman correlation value between Green and Blue channel	56	Histogram	kurt_hist_H	Kurtosis of the histogram of Hue channel
8	Colour-based	cor_GH	Spearman correlation value between Green and Hue channel	57	Histogram	skew_hist_H	Skewness deviation of the histogram of Hue channel
9	Colour-based	cor_GS	Spearman correlation value between Green and Saturation channel	58	Histogram	std_hist_S	Standard deviation of the histogram of Saturation channel
10	Colour-based	cor_GV	Spearman correlation value between Green and Value channel	59	Histogram	kurt_hist_S	Kurtosis of the histogram of Saturation channel
11	Colour-based	cor_GI	Spearman correlation value between Green and Intensity channel	60	Histogram	skew_hist_S	Skewness deviation of the histogram of Saturation channel
12	Colour-based	cor_BH	Spearman correlation value between Blue and Hue channel	61	Histogram	std_hist_V	Standard deviation of the histogram of Value channel
13	Colour-based	cor_BS	Spearman correlation value between Blue and Hue channel	62	Histogram	kurt_hist_V	Kurtosis of the histogram of Value channel
14	Colour-based	cor_BV	Spearman correlation value between Blue and Value channel	63	Histogram	skew_hist_V	Skewness deviation of the histogram of Value channel
15	Colour-based	cor_BI	Spearman correlation value between Blue and Intensity channel	64	Histogram	std_hist_R	Standard deviation of the histogram of Red channel
16	Colour-based	cor_HS	Spearman correlation value between Hue and Saturation channel	65	Histogram	kurt_hist_R	Kurtosis of the histogram of Red channel
17	Colour-based	cor_HV	Spearman correlation value between Hue and Value channel	66	Histogram	skew_hist_R	Skewness deviation of the histogram of Red channel
18	Colour-based	cor_HI	Spearman correlation value between Hue and Intensity channel	67	Histogram	std_hist_G	Standard deviation of the histogram of Green channel
19	Colour-based	cor_SV	Spearman correlation value between Saturation and Value channel	68	Histogram	kurt_hist_G	Kurtosis of the histogram of Green channel
20	Colour-based	cor_SI	Spearman correlation value between Saturation and Intensity channel	69	Histogram	skew_hist_G	Skewness deviation of the histogram of Green channel
21	Colour-based	cor_VI	Spearman correlation value between Value and Intensity channel	70	Histogram	std_hist_B	Standard deviation of the histogram of Blue channel
22	Colour-based	mean_H	Mean of the Hue channel	71	Histogram	kurt_hist_B	Kurtosis of the histogram of Blue channel
23	Colour-based	std_H	Standard deviation of the Hue channel	72	Histogram	skew_hist_B	Skewness deviation of the histogram of Blue channel
24	Colour-based	mean_S	Mean of the Saturation channel	73	Histogram	std_hist_I	Standard deviation of the histogram of Intensity channel
25	Colour-based	std_S	Standard deviation of the Saturation channel	74	Histogram	kurt_hist_I	Kurtosis of the histogram of Intensity channel
26	Colour-based	mean_V	Mean of the Value channel	75	Histogram	skew_hist_I	Skewness deviation of the histogram of Intensity channel
27	Colour-based	std_V	Standard deviation of the Value channel	76-85	Texture	lbp_[0-9]	Local Binary Pattern Vector
28	Colour-based	mean_R	Mean of the Red channel	86	Texture	com_entropy	Entropy of Co-occurence Matrix
29	Colour-based	std_R	Standard deviation of the Red channel	87	Texture	com_inertia	Inertia of Co-occurence Matrix
30	Colour-based	mean_G	Mean of the Green channel	88	Texture	com_energy	Energy of Co-occurence Matrix
31	Colour-based	std_G	Standard deviation of the Green channel	89	Texture	com_correlation	Correlation of Co-occurence Matrix
32	Colour-based	mean_B	Mean of the Blue channel	90	Texture	com_homogeniety	Homogeneity of Co-occurence Matrix
33	Colour-based	std_B	Standard deviation of the Blue channel	91	Texture	FFT_energy	Energy of FFT
34	Colour-based	mean_I	Mean of the Intensity channel	92	Texture	FFT_entropy	Entropy of FFT
35	Colour-based	std_I	Standard deviation of the Intensity channel	93	Texture	FFT_intertia	Inertia of FFT
36	Colour-based	entropy_I	Entropy of the Intensity channel	94	Texture	FFT_homogeneity	Hmogeneity of FFT
37	Image Quality	SNM	Statistical Naturalness Measure				

Table 5 – ML-based Performance

Meta-Model	Accuracy	Sensitivity	Specificity	Precision	Recall	F1	Balanced Accuracy
SVM	0.53	0.53	0.76	0.53	0.53	0.53	0.65
RF	**0.69**	**0.69**	**0.84**	**0.69**	**0.69**	**0.69**	**0.76**
XGBoost	0.61	0.61	0.80	0.61	0.61	0.61	0.71
Random	0.36	0.36	0.68	0.36	0.36	0.36	0.52
OneR	0.33	0.33	0.66	0.11	0.33	0.16	0.5

Table 6 – Gradient-based Performance

Meta-Model	Accuracy	Sensitivity	Specificity	Precision	Recall	F1	Balanced Accuracy
SVM	0.82	0.82	0.955	0.85	0.82	0.82	0.88
RF	**0.89**	**0.89**	**0.97**	**0.89**	**0.89**	**0.89**	**0.93**
XGBoost	0.77	0.77	0.94	0.76	0.77	0.76	0.85
Random	0.19	0.19	0.79	0.19	0.19	0.19	0.49
OneR	0.2	0.2	0.8	0.04	0.2	0.06	0.5

WORKS PUBLISHED BY THE AUTHOR

1. Aguiar, G.J.; Mantovani, R.G.; Mastelini, S.M.; de Carvalho, A. C. P. F. L.; Campos, G.F.C.; Barbon, S., **A meta-learning approach for selecting image segmentation algorithm**, Pattern Recognition Letters. (Qualis CC 2016, A1)

2. Aguiar, G. J.; Santana, E. J.; Mastelini, S. M.; Mantovani, R.G.; Barbon, S.. **Towards meta-learning for multi-target regression problems**. In 2019 8th Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2019. p. 377-382. (Qualis CC 2016, B2)

3. Campos, G.F.C.; Mastelini, S.M.; Aguiar, G.J.; Mantovani, R.G.; de Melo, L.F.; Barbon, S., **Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization**, EURASIP Journal on Image and Video Processing, 2019, p. 59, Springer, (Qualis CC 2016, B1)

Submited and Under review papers.

1. Aguiar, G. J.; Santana, E. J.; Guido, R.C.; Proença, M. L.; Barbon, S.. **Multi-label classification of voice disorders**, Computer Methods and Programs in Biomedicine. Submited. (Qualis CC 2016, A2)