Image analysis for automatic measurement of crustose lichens

Pedro Guedes, 73637, pedro.p.guedes@tecnico.ulisboa.pt

Abstract—Lichenometry is an exposure dating technique for rock surfaces, widely used in the study of geological deposits, which is based on the relationship between the size of lichens and their age.
Most of the published Lichenometry works apply traditional techniques to measuring lichen thalli that suffer from some problems such as lack of replicability and high variance of data collected by different operators. Even if using digital photography, the processing work is still non-automated, manual, time consuming and laborious process, especially when the number of samples is high.
This work developed a set of image acquisition and processing tools to efficiently identify lichens in rocky surface and produce relevant statistics (coverage percentage, number o individual lichens and area of each individual in mm^2).
The hardware component is composed of a digital camera and specially designed targets that allow the automatic image correction and scale assignment. The software allows the manual classification of images using a interactive foreground extraction (based on GrabCut) and the automatic image segmentation using SLIC (Simple Linear Iterative Clustering) and the SVC (Support vector machines) e Random Forest classifiers.
The initial evaluation shows promising results. With respect to manual image processing time the gains are higher than 75% when using the developed tool and with precision on the order of 95%. When using the automatic classifiers, the attained precision is higher than 70%. The developed system allows a reduction of lichen photograph images data set processing and shows greater potential for the automatic processing of such data.

Index Terms—ichenometry, lichens, machine learning, classification, segmentation.

1 Introduction

Lichenometry is a dating technique for rock surfaces, widely used in the study of geological deposits. It is based on the relationship between the size of lichens and their age. Alternative techniques for dating rock surfaces in geological studies, such as cosmogenic isotope dating, are time and resource consuming, especially if the number of individual samples is large. Lichenometry presents itself as a low-cost alternative for estimating the age of rock surface exposure.

There are several methods of collecting lichen dimensions for dating with lichenometry. The traditional method is composed of the following steps: selecting lichens, measuring manually, and writing down the collected parameters.

The dimensions of lichens have been measured directly on site using rulers, or on photographs [1], [2], [3]. A simple technique to estimate the diameter of a lichen thallus is to calculate the diameter of the largest inscribed circle, or the largest axis [4].

Technological advances in recent decades, particularly in the area of CMOS image sensors, allow access to high definition and low cost digital cameras. Eventually, with the evolution of, and easy access to, technology, certain procedures for measuring lichens have been done using digital images and with the aid of image editing software to measure individual sizes, axes, perimeter, and area [1]. The use of software like Adobe® Photoshop® or geographic information systems (e.g. ESRI® ArcGIS™), allows one to manually delineate the region of interest in the photograph in order to facilitate measurements of lichens and convert pixels into area [5], [6].

An example of the application of lichenometry is the study carried out on the rocky cliff coastline on the Portuguese coast, in particular on the limestone cliffs of Ericeira [6], for which there is a lichen growth model of the species *Opegrapha durieui* Mont [7]. These surfaces include cliff faces that have undergone dismantling and resulted in block falls and also coastal block deposits resulting from flooding by extreme marine events such as storms or tsunamis [6], [8].

Other studies make use of notable points in rocks (e.g. crystals with particular shape/dimension, fractures) to overlay images with different dates and to measure relative distances of growth within specific time periods, and determine direct lichen growth rates [1], [2].

Lichens are categorized according to certain characteristics, including the way they adhere to the substrate, the thallus, and the reproductive organs. Lichenometry has been essentially based on saxicolous crustose lichens (rocky substrate) with circular growth, such as the genus *Rhizocarpon* [9]. Crustose lichens develop perfectly attached to the substrate, their growth occurring essentially spread (flattened shape) along the boundaries of the thalli and marginally in thickness [10], [11]. Thus, two-dimensional measurement of their growth is best made using area growth [6].

Most published works about lichenometry apply traditional techniques for measuring lichen thalli [12], [13], [14], [15]. In these works the problem in using rulers or templates to make measurements is that they introduce errors and rough approximations in the obtained results [5], [6], [16]. Another aspect to keep in mind is that, with simple measuring instruments, it is not possible to measure the area directly. It is necessary to measure indicators (e.g. major/minor axis) which can become a time consuming task as well as only provide approximations (lichens are not perfectly cylindrical/elliptical and using only the dimension of axes, relevant information is lost). These classical data acquisition and processing techniques generate some problems, such as lack of replicability and high variance of data.

More recent studies of lichen cover analysis on rock have made use of digital cameras and dedicated image processing software to obtain their results and sample analysis [17]. However, this process remains a non-automated manual process, which makes it time consuming and labor intensive, especially when the number of samples is high. In addition, its dependence on decisions made during the procedure increases the variance of measurements made by more than one operator.

In this work we propose to develop image acquisition and processing tools and methodologies to solve the problems associated with data acquisition and processing for lichenometry application, namely the individualization and measurement of lichens. It is proposed to use digital image processing, computer vision and machine learning techniques in order to automate the segmentation and measurement of lichens in images, to produce methodologies and tools for image capture and processing for lichenometry. The aim is to automate the collection of information, reducing data collection time and facilitating field work (e.g. selection of thalli to be measured), to improve accuracy in the individualization and measurement of thalli (analysis and processing of collected data) and to create a methodology that can be replicated anywhere and by any user.

Due to the large variety of lichens and surfaces they colonize, one of the objectives of this work is the development of a generic application to process the data of several past and future field campaigns (without knowing the specific lichen species and surface of campaign), so that this system and methodologies that can be used in the future, in diverse campaigns with variability of lichen species and surface type. However, the system was developed for saxicolous (rocky substrate) crustose lichens and was tested only with these type of lichens.

The developed system is composed of a hardware and a software component. The hardware component consists of a digital camera and targets specially designed and evaluated in the scope of this work for this specific application. The software component consists of a series of algorithms and methodologies for processing the acquired samples (digital images) that produce the following results: segmentation of the lichen surfaces, percentage of coverage, counting the number of individuals and size of each individual (measured in mm^2).

The various components were evaluated in order to understand the benefits of their use when compared to non-automated methods and to measure possible errors. Image sets from 5 different locations with different lichen species and colonized surfaces were used in this evaluation.

2 Related work

2.1 Lichenometry

Lichenometry is a dating technique that is generally used on recently exposed rock surfaces (<500 years), where its use combines knowledge of biology (lichenology) and geology, and ecology. This technique is based on the relationship between the size of lichens and their age. If the growth rate of a given species is known, for example through predetermined growth models that relate size to age, then the time since exposure of the surface colonized by that lichen species can be inferred.

Building a robust lichen growth model requires measuring and evaluating its consistency in space, since the growth rate of lichens depends on several factors, such as species and climate, just to list a few [15], [18]. The growth curves representative of a region are obtained based on the diameter of lichens with known ages. Sampling for application in lichenometry requires identification of lichen species as well as estimation of the thallus, to characterize the population on the surface of interest.

2.2 Sampling techniques

The classical sampling method used in lichenometry was based on the selection and measurement, using standard instruments (e.g. ruler), of the major axis or diameter of the largest circumscribed circle of isolated lichen thalli (without visible coalescence) [9].

More recently, and with the purpose of preserving and increasing the quantity of samples collected in the field, measurements based on digital images started to be used. Based on these images, it was thus possible to later measure the diameter and areas of individual thalli or even the coverage areas of coalescing lichen thalli [6]. Regardless of the larger or lower complexity in image acquisition methods, image processing (lichen-rock differentiation and lichen delineation) has invariably been done manually or semi-automatically, using image software assisted selection tools.

The transformation from number of pixels to dimension requires the existence of elements with known dimension, for example including scales/rulers in the photographs [6]. This makes possible the extraction of the dimensions/area occupied by lichens.

2.3 Computer vision and classification techniques

2.3.1 *GrabCut*

Most segmentation techniques make use of information about regions and region boundaries contained in the image in order to perform segmentation. GrabCut [19] is a segmentation technique that uses both to perform the segmentation.

To perform the segmentation a graph is constructed, where the nodes of the graph represent pixels in the image. In addition, two special nodes are also created, the Sink and Source nodes marked by the user in the image as background and foreground correspondingly. Each pixel node in the graph is connected to the Source node and the Sink node. In order to segment the image, the Source and Sink nodes have to be separated.

For foreground and background regions to be created, some pixels in the image need to be classified by the user prior to segmentation as foreground or background. Any pixels that are classified during this phase are defined as constraints. This means that during the segmentation process, the classification of the pixels as foreground or background identified by the user do not change.

An energy/cost function is incorporated into the graph as weights between pixel nodes and weights between pixel and Source or Sink nodes. The weights between pixel nodes are determined by whether or not they are located in a border region in the image. Thus, a strong indication of a border between two pixels (a large difference in the color of the pixels) results in a very small weight between two pixel nodes (analogously, the weight between two pixel nodes with similar color will be high). The region information determines the weight between the pixel nodes and the Source and Sink nodes. These weights are calculated by determining the probability of the pixel node being part of the background or foreground region (this probability is given by a modeled Gaussian mixture model (GMM) [19] with the initial background and foreground inputs classified by the user.

A Min-cut/Max-Flow algorithm is used to segment the graph. This algorithm determines the minimum cost cut, determined by sum of all the weights of the links that are cut (minimization of the energy/cost function), which will separate the Source and Sink nodes. Once the Source and Sink nodes are separated, all pixel nodes connected to the Source node become part of the foreground, and the rest become part of the background.

In short, GrabCut is an image segmentation method based on graph cuts. Starting with a user-specified bounding box around the object to be segmented, the algorithm estimates the color distribution of the target object and that of the background, using GMM. This is used to construct a graph over the pixel classes, with an energy function that prefers regions connected with the same class, and running an optimization based on graph cuts to infer their values [20]:

- Estimate the foreground and background color distribution using a Gaussian Mixture Model (GMM).
- Building a graph over the pixel classes (i.e. foreground vs. background).
- Applying a graph cut optimization (Min-cut/Max-Flow algorithm) to arrive at the final segmentation.

The estimates can be further corrected by the user pointing out misclassified regions and running the optimization.

2.3.2 SVM and Random Forests

In machine learning, support-vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis [21]. Given a training data set consisting of already classified images, a SVM training algorithm builds a model that assigns new data to one class or another, making it a non-probabilistic binary linear classifier. The SVM maps training data to points in space and seeks to maximize the width of the interval between the points of the two classes (in the case of binary classification there are only two classes). New data is then mapped into that same space and predicted to belong to one of the two classes based on which side of the boundary they lie on. In addition to performing linear classification, SVM can efficiently perform non-linear classification using what is called the kernel trick by implicitly mapping its inputs into high-dimensional feature spaces.

Formally, a SVM constructs a hyperplane or set of hyperplanes in a high-dimensional space, which can be used for classification, regression, or other tasks such as outlier detection [22]. Intuitively, in the case of classification, good separation is achieved by the hyperplane that has the greatest distance to the nearest training data point of any class, since in general the larger the margin, the smaller the generalization error of the classifier [21].

While the original problem can be defined in a dimensionally finite space, it is often the case that the sets to be discriminated are not linearly separable in that space. For this reason, it has been proposed that the original space be mapped to a much higher space, presumably facilitating separation in that space [23]. To keep the computational load reasonable, the mappings used by the SVM schemes are designed to ensure that the scalar products of the input data vector pairs can be computed easily in terms of the variables in the original space by defining them in terms of a kernel function $k(x, y)$ selected to fit the [24] problem.

Random Forests is a particular type of learning algorithm based on decision trees. In decision tree learning, decision trees are used as predictive models to go from observations about an item (represented in the branches) to conclusions about the target value of the item (represented in the leaves).

Tree models where the target variable can take a discrete set of values, are called classification trees; in these tree structures, the leaves represent the class labels and the branches represent conjunctions of features that lead to those class labels.

Some techniques, often called ensemble methods, build more than one decision tree. Bootstrap aggregated (bagged) decision trees build multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction [25]. A random forest classifier is a specific type of bootstrap aggregating. Random forests correspond to an ensemble learning method for classification, regression and other tasks, which works by building a multiplicity of decision trees in the training period. For classification tasks, the result of random forests is the class selected by most trees [21].

2.3.3 Simple Linear Iterative Clustering (SLIC)

SLIC [26] creates super-pixels based on k-means clustering. Super-pixels are small regions of pixels in the image that share similar properties (color). Super-pixels simplify images with a large number of pixels making them easier to handle in many domains (computer vision, pattern recognition and machine learning). This algorithm generates superpixels by grouping pixels based on their color, similarity and proximity in the image plane.

The SLIC function from the python library skimage has the following parameters:

- n_segments: Approximate number of SLIC segments created for the input image.
- compactness : Balances color proximity and space proximity. Higher values give more weight to the proximity to space, making super-pixel shapes more square. This parameter strongly depends on the contrast of the image and the shapes of objects in the image.

- sigma: Width of the Gaussian smoothing kernel for the preprocessing of each dimension of the image. The same sigma is applied to each dimension in the case of a scalar value. Zero means no smoothing.

It was decided to divide the images into regions given by SLIC to train the classifiers. Preliminary tests carried out in the course of this work indicate that this approach provides good delimitation between lichens and rock, i.e. the boundary regions of the SLIC segments largely coincide with the boundaries between lichens and rock. This drastically reduces the number of features (number of features becomes equal to the number of SLIC segments instead of the total number of pixels).

2.4 Image processing applied to ecology

Several computer vision techniques have been applied in the context of vegetation analysis. These techniques generally involve image segmentation and classification with the interest of measuring certain geometric features such as leaf area, height or volume (to estimate growth), as well as identifying species or possible plants affected by disease [27], [28].

Some examples of the use of machine learning and image processing methods in the context of computer vision related to vegetation analysis include:

- Identifying the threshold in a binary segmentation problem to remove background in greyscale lettuce images using Otsu's method in order to automate lettuce area measurements [29] (in the field of computer vision and image processing, Otsu's method is used to automatically select the threshold of a greyscale image [30]). In its simplest form, the algorithm returns a single threshold intensity that separates the pixels into two classes, foreground and background;
- Identification of a vegetation index for lichens based on hyperspectral measurements (in the visible to mid-infrared spectrum) using samples as training and validation data sets to find the optimal values by minimizing the RMSE. Spectral mixing of lichens and rocks can make it difficult to diagnose materials of interest, thus leading to misinterpretations and false positives if mapping is done based on spectral matching methodologies. Therefore, the ability to distinguish lichen cover from rock and decompose a blended pixel into a collection of pure reflectance spectra may improve the applicability of hyperspectral methods for mineral exploration [31].

However, to date there is no solution that solves the sampling problems for lichenometry application.

3 Automating image processing

3.1 Description of the system from the user's point of view

The objectives of this project are related to the development of a tool for automatic measurement of lichen dimensions in order to contribute to better data acquisition for dating with lichenometry in geological and/or archaeological studies.

The user flow (Figure 1) consists of the following steps:

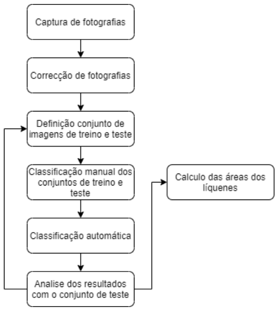

Captura de fotografias
Correcção de fotografias
Definição conjunto de imagens de treino e teste
Classificação manual dos conjuntos de treino e teste
Classificação automática
Analise dos resultados com o conjunto de teste
Calculo das áreas dos líquenes

Fig. 1. User flow.

- Photo Capture - The tool will be applied to previously sampled data sets or to new data sets acquired during the course of this project. It is up to the user to choose which data set to analyze using the program.
- Photo correction - Given the chosen data set, each photo in the data set is corrected for perspective errors.
- Definition of the set of training and test images - Definition of two subsets (training and test) of images belonging to the chosen data set. The training images serve to train the classifier to identify the regions of interest (lichens) and the test images serve to evaluate and benchmark the performance of the system.
- Manual classification of the training and test sets - The regions of interest (lichens) of the training and test images are explicitly identified by manually classifying these images with the help of a background extraction tool. This results in binary images with the lichen class regions in white and the rest in black. These images together, with the corresponding training and test images, allow the program to learn.
- Automatic classification - The program will automatically classify the regions of the data set images that are lichens.
- Analyzing results with the test set - The system returns the segmented images and the performance of the measurements made on the test set of images. Depending on the performance it may be necessary to re-run the program with more training images.
- Calculation of lichen areas - For each chosen segmentation the area of each individual lichen is measured.

The next sections describe each element present in Figure 1 in detail.

3.2 Photo capture and correction

One of the problems associated with photograph sampling, particularly relevant when the goal is to extract spatial information (such as the area occupied by a lichen thallus), is image deformation. This occurs either due to the oblique orientation of the photograph relative to the surface, or due to the deformation of the lens.

Thus, in the acquisition of new images in the field, we propose the use of 4 blue targets arranged over the vertices of a square/rectangular region that includes the area of interest to be photographed. This will allow corrections and transformations to be applied in order to compensate for perspective errors and assigns a scale.

The size (width and height) of the rectangular region enclosed by the targets must be recorded during sampling and later entered into the program so that image deformation can be removed.

In a first iteration of the target system, the detection of the Leica® topographic targets was made using SIFT [32] descriptors and using RANSAC. The SIFT generates descriptors that represent notable points in an image allowing matching the SIFT descriptors of the search image (image with only the topographic target to detect) to the SIFT descriptors of the distorted image (the matching will be done with one of the 4 targets in the image). This correspondence is done with RANSAC, knowing that a correspondence can only exist between points of the same plane (homography). Since all 4 targets were identical, whenever a correspondence was made between the search image and a target, the program had to cover that specific target from the image. Otherwise the correspondence would always be made between the same descriptors SIFT, belonging to the same target.

While this method produces some promising results in controlled experiments, it is not robust enough to perform target detection in images with feature-rich rocky backgrounds. More importantly, when the perspective deformation of the image is more intense this system can no longer detect all 4 targets and starts matching wrong points.

Robustness was improved using a simpler solution through color segmentation, using 4 circular targets with the same blue color. This color was chosen in order to provide a good contrast and because it is uncommon in both rock formations and lichens.

In order to determine the parameters necessary for color segmentation of the targets, the images photographed with the target system were first converted from RGB to HSV. This makes it easier to define the upper and lower limits of the color channels in order to segment the targets. The experimentally defined H, S and V ranges were: H[95-105], S[85-255] and V[170-245].

The system works by detecting the centers of mass of the segmented regions, corresponding to the targets. With the coordinates of these 4 points and knowing the real geometry of the targets (rectangle with 27.2cm length and 18.5cm height, for example) a geometric transformation matrix is calculated. With this matrix, the perspective of the image is transformed in order to eliminate errors from the original perspective and facilitate measurements.

Figure 2 shows that the deformation of the circular target in the center of the target region is corrected.

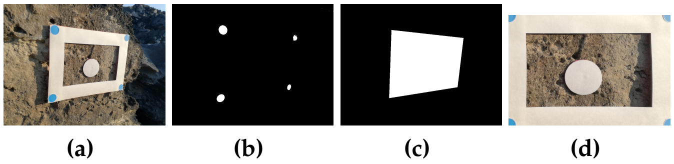

(a) **(b)** **(c)** **(d)**

Fig. 2. Image correction example: (a) original photograph, (b) target detection, (c) interest area crop, (d) final corrected image.

3.3 Manual classification process

The manual classification generates binary outputs corresponding to the training and test images, in which the regions belonging to the lichen class are identified in white and the others in black, allowing the user to ’teach’ the program which features of interest in the images. These binary images, together with the corresponding training and test images, serve as a reference for the program to learn.

The manual image classification process is done using the GrabCut algorithm. From the user’s point of view, the GrabCut algorithm works by accepting an input image where:

- the user identifies several areas corresponding to the lichens,
- the user identifies several areas corresponding to the background,
- the system updates the binary result and,
- if the resulting classification contains obvious errors, the user can identify additional lichens and background.

Sometimes the segmentation performed by the algorithm based on the user’s delimitation is far from ideal. In such cases, fine touches need to be made by selecting defective results and marking them properly.

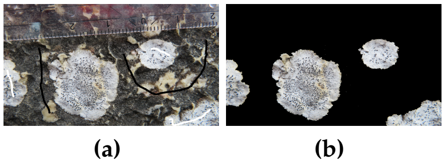

(a) **(b)**

Fig. 3. Example of foreground (white strokes) and background (black strokes) selection and final segmentation using GrabCut.

As can be seen in the Figure 3, after some final retouching, identified by the white (denoting foreground) and black (denoting background) strokes, a good segmentation result is obtained which, in this case, separates lichen from rock.

3.4 Automatic classification

In this section all the steps and operation of the components that perform the automatic classification are explained.

The automatic lichen classification program receives as input data the directory where the images are located (data set). The user has to define the number of pictures that will be used for training and also for testing (randomly chosen). These pictures will be manually classified as described in Section 3.3. These data sets allow the classifiers to be trained with properly classified data and provide a benchmark for evaluating the program’s automatic classification performance. The flow of the program is illustrated in Figure 4.

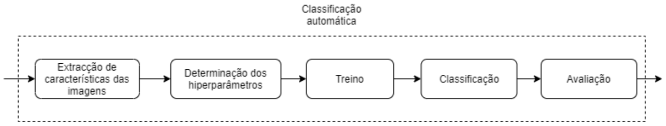

Classificação automática
Extracção de características das imagens
Determinação dos hiperparâmetros
Treino
Classificação
Avaliação

Fig. 4. Flow of the automatic classification.

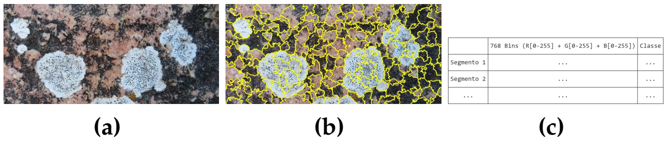
The automatic classification component receives the training and test images as well as the corresponding binary images manually classified with GrabCut. The output returns the segmentations of the remaining images in the data set (images that do not belong to the training and test sets).

The program uses as features the relative frequency histograms corresponding to the SLIC segments of the images. Alternatively, each pixel could be used as a feature for training and classification. The use of the SLIC segments allows speeding the execution of the training and classification since the number of training features is reduced (the number of pixels is much larger than the number of SLIC segments). It also allows preserving some local information of the image regions (fundamental for segmentation), since a pixel itself has no information about the surrounding region (unlike SLIC segments).

For each SLIC segment, the program will define a relative frequency histogram, representative of the pixels of that segment (in percentage of occurrence of each pixel of each color). Each relative frequency histogram serves as features to train the classifiers.

For each image, the program generates different sets of SLIC segments, each created with a different set of parameters. The range of SLIC parameters tested is: n_segments = [2000, 1000, 500], compactness = [20, 10], sigma = [3, 1], threshold = [0.5].

Thus, the training and test sets are converted into 12 different sets of training and test data corresponding to feature extraction with the 12 possible combinations of SLIC parameters. Each data set (training, test or singular image to segment), is therefore represented by a table where the rows correspond to the SLIC segments of all the images in the set and the columns correspond to the pixel color frequency histograms of each SLIC (row). In the case of the training and test sets, there is also a last column representing the class of each segment (lichen or background), product of the manual classification. Figure 5 shows the feature extraction for an image.



	768 Bins (R[0-255] + G[0-255] + B[0-255])	Classe
Segmento 1	...	...
Segmento 2	...	...
...	...	...

Fig. 5. Extraction of features from images.

Both the SVC classifier and the RandomForestClassifier have various combinations of parameters that can affect the classification performance:

- SVC - 'C': [1,10,100], 'kernel': ['rbf', 'linear', 'poly'], 'degree': [2,3,4,5], 'gamma': ['scale', 'auto'], 'max_iter': [500, 1000]
- RandomForestClassifier - 'n_estimators': [150, 100, 50], 'criterion': ['gini', 'entropy']

In order to find the best combination of parameters, it is necessary to make an initial evaluation usually called hyperparameter estimation.

For this, cross-validation using 5 folds was applied to the 12 sets derived from the SLIC parameters.

Training is done with the multiple combinations of hyperparameters of the classifiers and for each combination SLIC their performance is obtained.

The combination of parameters (for SVC and RandomForestClassifier) with the best performance is used in training and further classifications.

After determining the hyperparameters to be used by each of the classifiers, 24 classifiers are trained using the training set. These 24 versions correspond to instantiations of the SVC and RandomForestClassifier classifiers with the previously defined hyperparameters combined with 12 configuration alternatives of SLIC.

To evaluate the performance, the program segments the 12 test sets with the two classifiers thus producing 24 segmentations for each test set image.

The metric chosen to perform this evaluation was the Matthews correlation coefficient [33] (MCC). Each model segments the images in its test set and assigns to each a MCC value using manual classification as a reference. The performance of each classifier/parameter is obtained by averaging the MCC for the test set. This performance indicator for each of the 24 combinations of classifiers and parameters allows the to decide on adding more images to the training set and repeat the training and classification process; or choose one of the combinations and use the corresponding classification results.

Once trained and evaluated, it is possible to choose the classifier that can produce the best results based on the MCC calculated earlier. This classifier processes the remaining images in the data set and produces the corresponding classifications.

3.5 Calculation of lichen areas

The results of the classifications identify the areas occupied by lichens. However, it is essential that other data can be extracted, such as the number and area of lichens thalli.

These calculations are performed by a program that receives the binary images from the segmentations (black and white images) and, for each lichen, assigns an index, returns the area, filled_area (number of pixels in the region with all holes filled) perimeter and centroid coordinates.

The program uses the *regionprops* function from the *measure* library of *skimage* to perform the analysis of the image regions.

If the user assigns a scale to the image, it is possible to convert the area of each lichen from pixels to mm^2.

The scale can be taken automatically from the image if it was captured with the developed target system, or manually if the image includes another type of measuring instrument (ruler).

4 Evaluation

4.1 Description of data sets and test environment

All code was developed and tested on a machine with the following specifications: Windows 10, 64-bit, 8 GB of RAM, Intel(R) i7-2630QM CPU @ 2.00GHz 4 Cores CPU, Python 3.7.9, Opencv 4.5.1, Pandas 1.2.1, Scikit-learn 0.24.1, Scikit-image 0.17.2, Anaconda 1.9.12, Spyder 5.0.0.

Due to high execution times some tests (automatic classification and feature tests) were done on a different computer: Ubuntu 18.04.4, 15GB of RAM, Intel(R) Xeon(R) E3-1230 v5 @ 3.40GHz 8 Cores CPU.

The data sets present in Table 1 were used to evaluate the automatic classification program.

Name	Local	Nº imagens	Resolution
Terraço	Antartida	63	3888 x 5184
Nazaré 1	Nazaré	27	1944 x 2592
Nazaré 2	Nazaré	40	3456 x 4608
Nazaré 3	Nazaré	52	3456 x 5184
Muro Castelejo 2	Fundão	38	3456 x 5184
Muro Escola Castelejo	Fundão	17	3456 x 5184
Cascais	Cabo Raso	63	2000 x 3008

TABLE 1
Data sets used.

4.2 Targets system

In order to evaluate the targets system, experiments were performed with 18 photographs taken with the special mark. The experiments were performed at the same place where the Cascais data set was acquired (Table 1). The procedure is similar to what would be done to capture lichens, but in this case a special mark is used. Each experiment consists of: i) placing the special mark on a rocky surface, ii) placing the targets on the same surface so that they surround the mark, iii) taking a picture of the targets and mark, iv) correction of the image, v) measuring the major and minor axes of the mark, and vi) comparing the results to the reference values.

Both the targets and the mark have well-defined dimensions. The targets correspond to 4 blue circles glued onto a rectangular base, the centers of these targets form a rectangle 272mm long and 185mm wide. The special mark is a white circle with a 60mm diameter. The mark and targets are fixed to the rock surfaces with adhesive paste.

The measurements of the minor and major axes of the mark in the corrected image (Figure 6(b)) were performed automatically using the major_axis_length and minor_axis_length parameters of the *regionprops* function belonging to the python skimage library (Figure 6(c)).

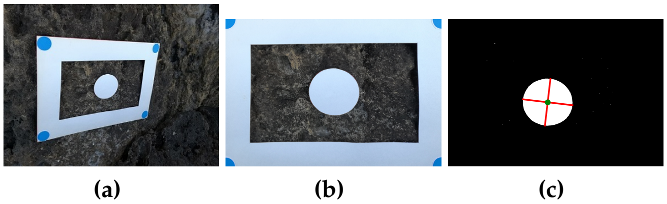

(a) **(b)** **(c)**

Fig. 6. Example of measuring the major/minor axes of the mark.

In one experiment it was not possible to detect the 4 targets, due to the presence of sky in the image, with a similar hue to the targets.

The average error obtained was 7.26%, and was from the fact that the target and the mark are not exactly in the same plane, since rocks surfaces ares irregular. As well as possible measurement errors of the mark radius and distances between the targets.

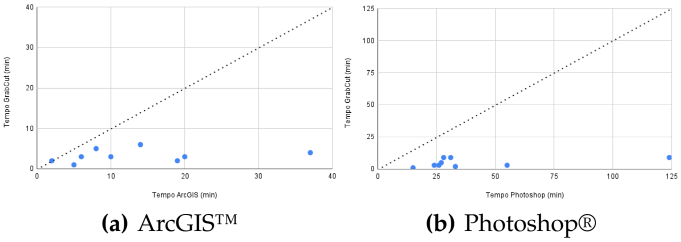
4.3 Image Processing

4.3.1 Manual classification

The use of GrabCut allows the user to reduce the time spent on image classification. This new tool is used for the creation of the training/test image sets, but can also be used alone, replacing the processing that uses other commercial software.

The evaluation of GrabCut's performance was performed by comparing ArcGIS™ and Photoshop® classifications as a reference, focusing on both time and classification. To perform this evaluation, a researcher was asked to classify 18 photographs: 9 were classified with Photoshop® and GrabCut and another 9 were classified with ArcGIS™ and GrabCut.

It can be seen (Figure 7) that GrabCut is always faster than the methods using Photoshop® and ArcGIS™. Furthermore, in a series of experiments, a 90% gain was obtained. In other words, in these cases the user only needs to spend 10% of the time that would have been spent using commercial software.



(a) ArcGIS™ **(b)** Photoshop®

Fig. 7. Time comparison between GrabCut and manual methods.

The associated errors obtained for all experiments were analyzed by comparing the GrabCut results against the results obtained with Photoshop® and ArcGIS™, chosen as references. This comparison was made in terms of MCC.

A mean MCC of 0.9174 was determined (y-axis of Figure 8). In order to analyze the MCC results in more detail, the relationship between MCC and the ArcGIS™/Photoshop® manual classification time is shown in Figure 8. It can be seen that there is a trend, and that the more complex/difficult is to segment an image, the greater the error of the classification performed with GrabCut. So, even in complex images, for which manual classification took approximately 2 hours, the GrabCut classification was under 10 minutes and the MCC was still above 0.75.

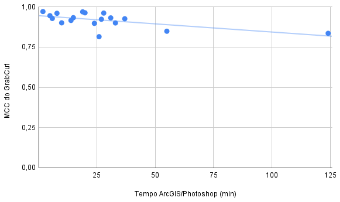


Fig. 8. Evolution of GrabCut error with manual classification time.

4.3.2 Characteristic evaluation (SLIC and histograms)

Since SLIC does clustering of areas in the images, it introduces errors, as the boundaries of the regions given by SLIC may not completely coincide with the lichen borders. Meaning that a given SLIC segment may contain both background and lichen pixels. Due to this possibility, an evaluation was be performed to quantify the error associated with the SLIC.

This evaluation was based on the same 18 images from the previous section and the corresponding (manually performed) binary classifications. However, they were downscaled to different resolutions (40%, 35%, 30%, 25%, 20%, 15% and 10%) from the original images in order to speed up the evaluation process and to study the temporal scalability of these features.

The SLIC parameters tested were the same as those used in the automatic classification: n_segments = [2000, 1000, 500], compactness = [20, 10], sigma = [3, 1], threshold = [0.5].

All 18 images were tested for the 12 parameter combinations SLIC, i.e. 216 tests and these tests were repeated for the same images but at different resolutions.

In each test, the SLIC clustering was applied to produce a set of segments. Each segment was assigned one class with help of the corresponding classification done with Photoshop®/ArcGIS™.

It is important to remember that the threshold parameter of the SLIC enters in the class assignment to each segment. For example, for a value of 0.5 (50%), when a given segment contains more than half of lichen pixels (you can tell by comparing with the classification done manually), then the lichen class is assigned to that segment, otherwise the background class is assigned.

Finally, the differences were compared and the times were measured. Feature generation times are the sum of the time to create the SLIC segments, the time to convert each SLIC segment into the corresponding relative frequency histogram (according to the colors of the pixels in that region) and, in the case of training data, the time to assign the class to each segment.

The average values of precision and of the MCC attained are relatively constant for the different image resolution, and the precision always in the order of 98% and MCC of 0.87. In cases where the size of the segments (given by the parameter n_segments) is larger than the lichen regions, the program may not be able to convert the lichen regions in the images to their features, originating feature segments assignment only to the background class. The precision value in these cases may be high (corresponding to the percentage of the background present in the image), however the MCC metric detects these cases resulting in MCC=0.

The experiments also show that the times depend exponentially with image resolution since, the higher the resolution of the images (more pixels to be processed), the longer the time it takes to calculate the histograms. And the higher the number of SLIC segments is, the longer it takes to calculate the histograms (each SLIC segment is converted into a corresponding histogram). It should be noted that the calculation of the histograms is the most time consuming process.

However, the most important thing to consider in these results is the error of the conversion to SLIC segments. We want to minimize this error since these features based on SLIC segments are used to train the classifiers. This error is due to the fact that the borders of the SLIC regions do not always coincide with the lichen borders. In the case of the tests performed on the set of images reduced to 30%, it was found that, in general, the error is low (high MCC values) and that this particular distribution presents a mean of 0.8679 and a standard deviation of 0.1327.

4.4 Classification results

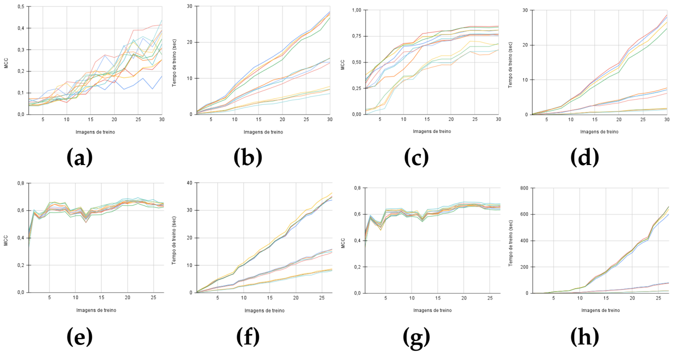
4.4.1 *Learning curves and system scalability*

In order to measure the learning capacity and scalability of the algorithms, several training cycles were performed for each of the 7 sets of images, each with an incremental number of training images. At the end of each training cycle, the images from the test set are segmented and compared to the reference set (manual classifications with GrabCut) to verify the cycle quality of the segmentations (with the metric MCC). Scalability was studied by analyzing the execution time of each cycle.

As described in the previous sections, in each training cycle, the two classifiers (RandomForestsClassifier and SVC were trained for each combination of the 12 SLIC parameters, resulting in 24 training cycles. Due to the characteristics of this evaluation, which implies the execution of several training cycles and that in each one the number of training images is increased (which further increases execution times) it was decided to exclude cross-validation. This was necessary to keep execution times at reasonable values.

The default parameters used for RandomForestsClassifier were: ’n_estimators’: [100], ’criterion’: [’gini’],and for the SVC were: ’C’: [1], ’kernel’: [’rbf’], ’gamma’: [’scale’], ’max_iter’: [-1]

Figure 9 illustrates the results obtained for the Antarctica and Cascais image sets. Each graph holds information for the 12 possible combinations of SLIC parameters for an individual classifier.



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

(e) **(f)** **(g)** **(h)**

Fig. 9. Learning and scalability curves for Antarctica (a, b, c, d) and Cascais (e, f, g, h) datasets.

Figure 9 shows that the learning curves tend to increase with the increase in the number of images used in each training cycle. meaning, the more images are used in training, the better the quality of the classifications.

Comparing the various learning curves shows that, depending on the set of images, the SLIC parameters may or may not impact the quality of the segmentations. This is due to the intrinsic variance in each set of images. It is also important to note that the classifiers may have differences in the quality of the segmentations , depending on the image set.. In general, SVC seems to show better results than the RandomForestsClassifier.

Similarly, the training time for both classifiers increases with the number of training images, i.e., the more images used, the longer it takes to perform the training.

Training times show similarities in the learning curves sharing the same parameter SLIC n_segments. This means that the training time is strongly linked to this parameter and that the more segments SLIC were generated, the greater the number of features in the training data, which forces the classifiers to process more information, increasing the training time and worsening the scalability of the algorithms.

It can also be seen that the training times of RandomForestsClassifier classifiers evolve linearly with time unlike SVC that evolve exponentially. We can also see that, for the Cascais image set, the SVC classifier has longer training times than the RandomForestsClassifier. Therefore, the RandomForestsClassifier presents better scalability than the SVC.

4.4.2 Segmentation analysis

In this section we analyze the segmentations to illustrate some problems and limitations of the developed program , as well as the potential related to this type of segmentation.

Some segmentations were chosen for analysis to illustrate the characteristics previously enunciated. The goal was to evaluate the quality of the produced segmentations, specifically with regard to true/false positive/negative.

Following the previous section, some segmentation of the Antarctica, Cascais and Muro Escola Castelejo sets (Figure 10 are presented. The images are arranged so that, in each row, we first observe the original image, then the manual classification and finally an automatic classification. The automatic classifications of the Figure 10(c) and Figure 10(f) (Antarctica) were produced using the SVC classifier, trained with 30 images and with the following SLIC parameters: threshold 0.5, n_segments 500, compactness 20 and sigma 1.

The automatic classification of Figure 10(i) (Cascais) was produced using the RandomForestsClassifier classifier, trained with 27 images and with the following parameters: threshold 0.5, n_segments 2000, compactness 10 and sigma 3.

The automatic classification of Figure 10(l) (Muro Escola Castelejo) was produced using the SVC classifier, trained with 27 images and with the following parameters: threshold 0.5, n_segments 500, compactness 20 and sigma 1.

For the first group of 3 pictures (Antarctica), it turns out that a perfect segmentation is obtained. However, it is important to note that, for exactly the same conditions but different image (Figure 10(c)), that same classifier produces a completely black image without the detection of the lichen. This is an example of a false negative.

For the Cascais set (Figure 9(e) and Figure 9(g)), it can be seen that, regardless of the classifier and the SLIC parameters, the segmentations are all, with rare exceptions, of poor quality. In fact, in Figure 10(i), several false positives are observed.

In the case of Figure 10(k), the classification was even better than the GrabCut manual classification.

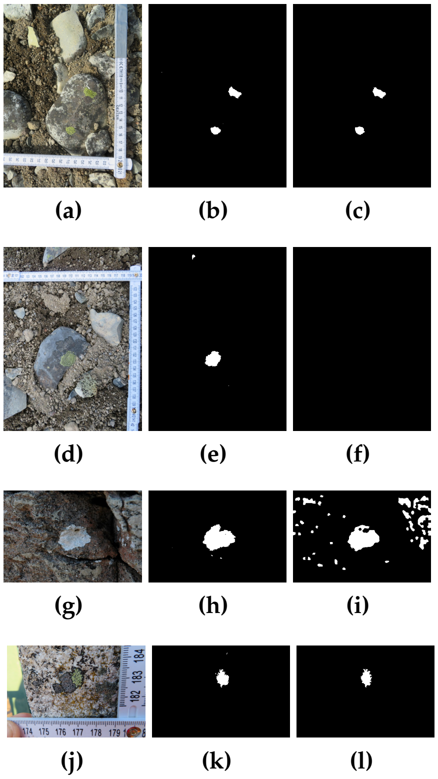


Fig. 10. Automatic classification for Antarctica (a, d), Cascais (g) and Muro Escola Castelejo (j).

5 Conclusion

A set of tools was produced to assist the acquisition and processing of lichenometric photographic samples. Even without considering the automatic classification, it was found that both the manual classification tool and the automated system of acquisitions with targets, improve and speed up the process. Estimates of lichen cover areas on rocks are automatically obtained for a larger number of data points with lower user time. This result makes more robust statistical analyses possible, and will contribute to studies that elaborate lichen growth models for age estimation using lichenometry.

With regard to manual classification, it was observed that large gains were obtained in classification times. These gains are due to the fact that the GrabCut algorithm considers both color and proximity between pixels to perform the segmentation. It was also observed that the quality of the classifications is on par with the classifications produced with the other methods.

The automatic classifier was developed to work for unknown future image sets sets, so ideal parameters must can not previously fixed. The classifier and feature generation parameters should be defined on a per data set case, by the user or using automatic parameter estimation.

The automatic classification does not performs optimally with data sets containing images with a wide scale range (that affect the SLIC algorithm), containing shades or where taken with various light intensity. This can be solved with field guidelines for the user and with future additional components to estimate more suitable parameters.

For future studies, solutions based on non-visible spectrum bands or even visible multi-spectral images can also be explored.

References

[1] N. M. Henry, “Measurement of growth in the lichen Rhizocarpon geographicum using a new photographic technique,” Master’s thesis, Faculty of Mathematics and Science, Brock University St. Catharines, Ontario, April 2011.

[2] T. A. Bukovics, “Photogrammetric Exploration of Demographic Change in Juvenile Rhizocarpon geographicum Thalli,” Master’s thesis, Faculty of Mathematics and Sciences, Brock University St. Catharines, Ontario, 2016.

[3] D. P. McCarthy, “A simple test of lichenometric dating using bidecadal growth of rhizocarpon geographicum agg. and structure-from-motion photogrammetry,” *Geomorphology*, vol. 385, p. 107736, 2021.

[4] W. Locke and Et al., “A manual for lichenometry. british gcomorphological research group,” *Technical Bulletin*, vol. 26, pp. 1–47, 01 1979.

[5] D. P. McCarthy and K. Zaniewski, “Digital analysis of lichen cover: A technique for use in lichenometry and licnenology,” *Arctic, Antarctic, and Alpine Research*, vol. 33, no. 1, pp. 107–113, 2001.

[6] M. A. Oliveira and Et al., “Estimating the age and mechanism of boulder transport related with extreme waves using lichenometry,” *Progress in Physical Geography*, 2020.

[7] C. Roux and J. Egea, “L’opegraphetum durieui egea et roux ass. nov., une association lichénique saxicole-calcicole, halophile.” *Cryptogamie, Bryologie, lichénologie*, vol. 13, pp. 105–115, 1992.

[8] M. A. Oliveira, “Boulder deposits related to extreme marine events in the western coast of portugal.” Ph.D. dissertation, Universidade de Lisboa, Portugal, 2017.

[9] J. L. Innes, “Lichenometry,” *Progress in Physical Geography: Earth and Environment*, vol. 9, no. 2, pp. 187–254, 1985.

[10] D. Hill, “The growth of lichens with special reference to the modelling of circular thalli,” *The Lichenologist*, vol. 13, no. 3, p. 265–287, 1981.

[11] A. Seminara and Et al., “A universal growth limit for circular lichens,” *Journal of The Royal Society Interface*, vol. 15, p. 20180063, 06 2018.

[12] J. A. MATTHEWS, “Families of lichenometric dating curves from the storbreen gletschervorfeld, jotunheimen, norway,” *Norsk Geografisk Tidsskrift - Norwegian Journal of Geography*, vol. 28, no. 4, pp. 215–235, 1974.

[13] J. L. Innes, “Dating exposed rock surfaces in the arctic by lichenometry: The problem of thallus circularity and its effect on measurement errors,” *Arctic*, vol. 39, no. 3, pp. 253–259, 1986.

[14] D. P. McCarthy, “Estimating Lichenometric Ages by Direct and Indirect Measurement of Radial Growth: A Case Study of Rhizocarpon agg. at the Illecillewaet Glacier, British Columbia,” *Arctic, Antarctic, and Alpine Research*, vol. 35, no. 2, pp. 203 – 213, 2003.

[15] J. A. Matthews and H. E. Trenbirth, “Growth rate of a very large crustose lichen (rhizocarpon subgenus) and its implications for lichenometry,” *Geografiska Annaler: Series A, Physical Geography*, vol. 93, no. 1, pp. 27–39, 2011.

[16] H. E. Trenbirth, “Lichenometry.” Geomorphological Techniques (Online Edition). British Society for Geomorphology; London, UK. ISSN: 2047-0371., 2010, ch. Section 4.2.7.

[17] C. Gazzano and Et al., “Image analysis for measuring lichen colonization on and within stonework,” *The Lichenologist, British Lichen Society*, vol. 41, no. 3, p. 299–313, May 2009.

[18] R. Armstrong, *The influence of environmental factors on the growth of lichens in the field*. Germany: Springer, 2015, vol. 1, pp. 1–18.

[19] C. Rother and Et al., “"grabcut": Interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, vol. 23, no. 3, p. 309–314, Aug. 2004.

[20] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary region segmentation of objects in n-d images,” in *Proceedings Eighth IEEE Intl. Conference on Computer Vision. ICCV 2001*, vol. 1, 2001, pp. 105–112 vol.1.

[21] T. Hastie and Et al., *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2008.

[22] “1.4 Support Vector Machines, scikit-learn 0.20.2 documentation,” 2020. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html

[23] B. E. Boser and Et al., “A training algorithm for optimal margin classifiers,” in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

[24] W. Press and Et al., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007, vol. Section 16.5. Support Vector Machines.

[25] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug 1996.

[26] R. Achanta and Et al., “Slic superpixels,” *Technical report, EPFL*, 06 2010.

[27] M. Kırcı and Et al., “Vegetation measurement using image processing methods,” in *2014 The Third Intl. Conference on Agro-Geoinformatics*, 2014, pp. 1–5.

[28] J. Barbedo, “Digital image processing techniques for detecting, quantifying and classifying plant diseases,” *SpringerPlus*, vol. 2, p. 660, 12 2013.

[29] C. J. S. H. Lin Kaiyan, Wu JunHui, “Measurement of plant leaf area based on computer vision,” in *2014 Sixth Intl. Conference on Measuring Technology and Mechatronics Automation*, 2014ffff.

[30] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[31] S. Salehi and Et al., “Identification of a robust lichen index for the deconvolution of lichen and rock mixtures using pattern search algorithm (case study: Greenland),” *ISPRS - Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 973–979, 06 2016.

[32] D. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE Intl. Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[33] B. Matthews, “Comparison of the predicted and observed secondary structure of t4 phage lysozyme,” *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.