



Análise de imagem para medição automática de líquenes crustosos

Pedro António Pinto Parada Guedes

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientadores: Prof. João Nuno De Oliveira e Silva
Maria Alexandra Oliveira, PhD

Júri

Presidente: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Orientador: Prof. João Nuno De Oliveira e Silva
Vogal: Prof. Paulo Luís Serras Lobato Correia

Setembro 2021

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Agradecimentos

Gostaria de agradecer aos meus pais pela sua amizade, incentivo e carinho ao longo de todos estes anos, por sempre estarem lá para mim e sem os quais este projecto não seria possível.

Gostaria também de agradecer aos meus orientadores da dissertação, Prof. João Nuno De Oliveira e Maria Alexandra Oliveira, PhD, pela sua compreensão, apoio e partilha de conhecimentos que tornaram possível esta tese.

Por último, mas não menos importante, a todos os meus amigos e colegas que me ajudaram a crescer como pessoa e sempre estiveram lá para mim durante os bons e os maus momentos da minha vida.

Para todos e cada um de vocês - Obrigado.

Resumo

A liquenometria é uma técnica de datação de exposição de superfícies rochosas, amplamente utilizada no estudo de depósitos geológicos, que se baseia na relação entre a dimensão dos líquenes e a sua idade. A maioria dos trabalhos de liquenometria publicados, aplicam técnicas tradicionais de medição dos talos de líquenes que apresentam alguns problemas, como sejam a ausência de replicabilidade, elevada variância dos dados colhidos por diferentes operadores, etc. Mesmo com o uso de fotografia digital, esse processo continua a ser um processo manual, não automatizado, demorado e trabalhoso, especialmente quando o número de amostras é elevado.

Neste trabalho desenvolveu-se uma ferramenta de aquisição e processamento de imagem que permite a identificação de líquenes em rochas e cálculo de estatísticas (percentagem de cobertura, contagem do número de indivíduos e tamanho de cada indivíduo em mm^2). O hardware consiste numa câmara digital e miras especialmente definidas, que permitem a correção e atribuição de escala de forma automática. A componente de software consiste num sistema para classificação manual das imagens usando o algoritmo GrabCUT, e para a classificação/segmentação automática das imagens integrando o algoritmo SLIC (Simple Linear Iterative Clustering) com os classificadores SVC (Support vector machines) e Random Forest.

Os resultados são promissores. No que se refere ao processamento manual das imagens, obtêm-se ganhos superiores a 75% com precisão superior a 95%. O uso da classificação automática, consegue atingir precisão média superior a 70%. Os resultados de trabalho permitem uma agilização de processos até aqui muito lentos, tendo também grande potencia de automatização com precisão elevada.

Palavras Chave

liquenometria, líquenes, aprendizagem automática, classificação, segmentação.

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 stems 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 of 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 an 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 photographs images data-set processing and show greater potential for the automatic processing of such data.

Keywords

lichenometry, lichens, machine learning, classification, segmentation.

Conteúdo

1	Introdução	1
1.1	Enquadramento	2
1.2	Problema	3
1.3	Proposta de solução e objectivos	4
1.4	Estrutura do documento	4
2	Trabalho relacionado	6
2.1	Liquenometria	7
2.1.1	Técnicas de amostragem	9
2.2	Teoria de visão computacional e classificação	11
2.3	Técnicas de visão computacional e classificação	13
2.3.1	Classificação manual com GrabCut	13
2.3.2	Classificação automática com SVM e Random Forests	15
2.3.3	Validação cruzada para estimar hiperparâmetros	16
2.3.4	Simple Linear Iterative Clustering (SLIC)	18
2.3.5	Matthews correlation coefficient (MCC)	19
2.4	Processamento de imagens aplicado à ecologia	20
3	Automatização do processamento de imagens	22
3.1	Captura de fotografias	24
3.2	Correcção de fotografias	25
3.3	Processo de classificação manual	26
3.4	Classificação automática	27
3.4.1	Extracção de características das imagens	28
3.4.2	Determinação dos hiperparâmetros dos classificadores	29
3.4.3	Treino, classificação e avaliação de desempenho	29
3.4.4	Processo de escolha de classificador e parâmetros	30
3.5	Calculo das áreas dos líquenes	34

4	Avaliação	36
4.1	Descrição dos conjuntos de dados e ambiente de teste	37
4.2	Sistema de miras	38
4.3	Processamento de imagens	40
4.3.1	Classificação manual	40
4.3.2	Avaliação das características (SLIC e histogramas)	41
4.4	Resultados das classificações	44
4.4.1	Curvas de aprendizagem e escalabilidade do sistema	44
4.4.2	Análise das segmentações	45
5	Discussão	49
6	Conclusão	53
A	Anexos	60

Lista de Figuras

1.1	Exemplos de medições do comprimento e maior círculo inscrito em líquenes. [1]	2
2.1	Comparação de curvas de datação liquenométricas do Alasca e arredores. [2]	7
2.2	Exemplo de coalescência entre líquenes.	8
2.3	Imagem de líquenes da espécie <i>O. durieui</i> capturada no forte S Miguel de Arcanjo, 25-01-2020, no âmbito de um estudo de análise de depósitos geológicos [3].	9
2.4	Exemplo de vectorização de um líquen com recurso à aplicação Esri® ArcMap™ e seleção de pixéis de líquenes com Adobe® Photoshop®. Imagens cedidas por M.A. Oliveira.	10
2.5	Exemplo da segmentação 2D de uma imagem 3x3 [4].	14
2.6	Diagrama de validação cruzada [5].	17
2.7	<i>Stratified k-fold</i> exemplo com 4 <i>folds</i> e 3 classes [5].	17
2.8	Exemplo das regiões dos segmentos <i>Simple Linear Iterative Clustering</i> (SLIC) para os parâmetros: <i>n_segments</i> =250, <i>compactness</i> =10 e <i>sigma</i> =1.	18
3.1	Fluxo do utilizador.	23
3.2	Exemplos de diferentes tipos de aquisição.	25
3.3	Exemplificação de erro na detecção da terceira mira, do sistema com <i>Scale-invariant feature transform</i> (SIFT) e miras Leica®.	26
3.4	Exemplificação do funcionamento do sistema com segmentação de cor e miras azuis.	26
3.5	Exemplo da selecção de primeiro plano (traços brancos) e de segundo plano (traços pretos) e classificação final com recurso ao GrabCut.	27
3.6	Fluxo da classificação automática.	28
3.7	Extracção de características das imagens.	29
3.8	Resposta da primeira camada convolucional da rede VGG16 (pesos pré-treinados na <i>imagenet</i>) a uma imagem de líquen.	32

3.9	Exemplo de medição de líquenes numa imagem sem escala, os eixos representam a altura e largura da imagem em pixeis.	35
4.1	Exemplo de medição dos eixos maior/menor do alvo.	38
4.2	Caso de falha da detecção das miras.	39
4.3	Evolução do erro do GrabCut com o tempo de classificação manual do ArcGIS™ e Photoshop®.	41
4.4	Relação entre os tempos de execução totais (médios) e a resolução das imagens.	43
4.5	Distribuição dos resultados do <i>Matthews correlation coefficient</i> (MCC) para imagens com resolução 1555 x 1036.	44
4.6	Gráficos das curvas de aprendizagem e escalabilidade para os conjuntos de imagens da Antártida e Cascais.	46
4.7	Exemplos de classificações automáticas de imagens seleccionadas dos conjuntos da Antártica, Cascais e Muro Escola Castelejo.	48
A.1	Exemplos de duas imagens de cada conjunto de dados.	63
A.2	Curvas dos restantes conjuntos de dados.	64

Lista de Tabelas

3.1	Combinação dos parâmetros seleccionados classificadores e SLIC.	30
3.2	Mini-conjuntos de dados utilizados no desenvolvimento do programa.	31
3.3	Detalhes dos resultados do teste nº9 para cada uma das três imagens do conjunto de teste.	34
3.4	Resultado das medições para a Figura 3.9(a).	35
4.1	Especificações do hardware e ambiente utilizados no desenvolvimento do trabalho e testes.	37
4.2	Conjunto de dados usados.	37
4.3	Resultados das experiências efectuadas com as miras.	39
4.4	Tempos e erros medidos na avaliação das segmentações manuais.	40
4.5	Resumo dos resultados médios obtidos nos testes das características SLIC e histogramas para imagens de resoluções diferentes.	42
A.1	Tabela com os resultados dos melhores cinco pares classificadores-características dos nove testes para selecção e avaliação do modelo e características.	61
A.2	Tabela com os resultados dos melhores cinco pares classificadores-características SLIC dos nove testes para selecção e avaliação dos modelos e características SLIC.	62

Acrónimos

RGB	<i>Red, green, blue</i>
HSV	<i>Hue, saturation, value</i>
RANSAC	<i>Random sample consensus</i>
SIFT	<i>Scale-invariant feature transform</i>
CNN	<i>Convolutional neural network</i>
SVM	<i>Support vector machine</i>
SVC	<i>Support vector classifier</i>
RMSE	<i>Root mean squared error</i>
GMM	<i>Gaussian mixture model</i>
MCC	<i>Matthews correlation coefficient</i>
SLIC	<i>Simple Linear Iterative Clustering</i>
CV	<i>Cross validation</i>

1

Introdução

Conteúdo

1.1 Enquadramento	2
1.2 Problema	3
1.3 Proposta de solução e objectivos	4
1.4 Estrutura do documento	4

1.1 Enquadramento

A liquenometria é uma técnica de datação de exposição de superfícies rochosas, amplamente utilizada no estudo de depósitos geológicos, que se baseia na relação entre a dimensão dos líquenes e a sua idade. Técnicas alternativas de datação de superfícies rochosas em estudos geológicos, como datação com isótopos cosmogénicos, levam tempo e consomem recursos, especialmente se o número de amostras individuais for elevado. Assim, a liquenometria apresenta-se como uma alternativa de baixo custo para estimar a idade da exposição de superfícies rochosas.

Existem diversos métodos de recolha das dimensões dos líquenes para datação com liquenometria, que são aplicados dependendo dos objectivos. Apesar de existirem várias metodologias empregues, nesta secção serão apenas enumeradas algumas das mais comuns e utilizadas em estudos passados: métodos tradicionais e métodos electrónicos.

O método tradicional é composto dos seguintes passos: seleccionar os líquenes, medir manualmente e apontar os parâmetros recolhidos. Historicamente, dimensões de líquenes para utilização em liquenometria têm sido medidas directamente no local com recurso a réguas e craveiras (medição de eixo maior ou menor dos líquenes), assim como com aplicação de técnicas de fotografia [2, 6, 7]. Uma técnica simples para estimar o diâmetro de um líquen (um indivíduo) é calcular o diâmetro do maior círculo inscrito no talo do líquen [8] como se observa na Figura 1.1.

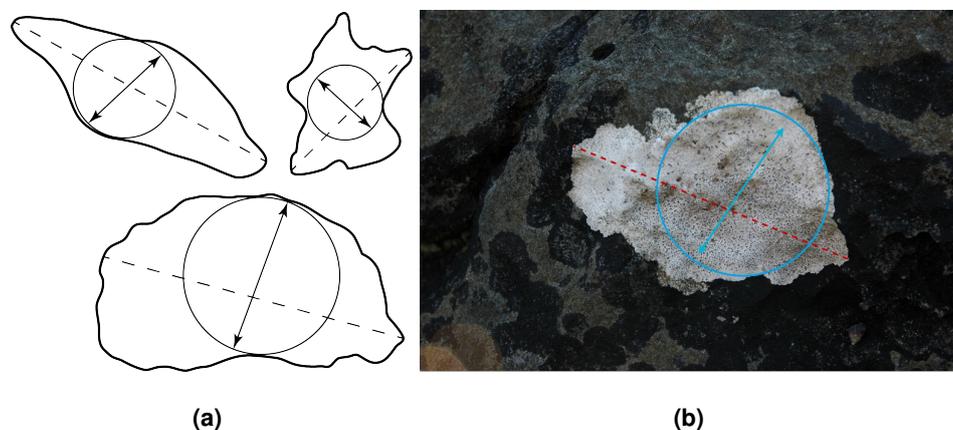


Figura 1.1: Exemplos de medições do comprimento e maior círculo inscrito em líquenes. [1]

Os avanços tecnológicos das últimas décadas, nomeadamente na área dos sensores de imagem CMOS, permitem acesso a câmaras digitais de elevada definição e baixo custo. Eventualmente, com a evolução da, e fácil acesso à, tecnologia, certos procedimentos para medição de líquenes têm sido feitos com recurso a câmaras digitais e com o auxílio de software de edição de imagem para medir os tamanhos dos indivíduos, eixos, perímetro e área [2]. O uso do Adobe® Photoshop® ou de software de sistemas de informação geográfica (por exemplo, ESRI® ArcGIS™), permite delimitar a região de

interesse na fotografia manualmente, de modo a facilitar medições dos líquenes e converter pixels em área [3, 9].

Um exemplo da aplicação de liquenometria é o estudo efectuado no litoral em arriba rochosa na costa Portuguesa, em particular nas arribas calcárias da Ericeira [3], para as quais existe um modelo de crescimento dos líquenes da espécie *Opegrapha durieui* Mont. (Roux and Egea, 1992 [10]). Estas superfícies incluem faces de arribas que sofreram desmonte e resultaram em quedas de blocos e ainda depósitos de blocos costeiros resultantes de inundações por eventos marinhos extremos (tempestades ou tsunamis) [1, 3].

Outros estudos fazem uso de pontos notáveis nas rochas (por exemplo cristais com forma/dimensão particular, fracturas) para sobrepor imagens com diferentes datas ou para medir distâncias relativas de crescimento [2, 6].

Os líquenes dividem-se de acordo com algumas características, entre elas, a forma como se aderem ao substrato, o talo e os órgãos reprodutivos. A liquenometria tem sido essencialmente baseada em líquenes crustosos saxílicos (substrato rochoso) com crescimento circular, tal como o género *Rhizocarpon* [11]. Os líquenes crustosos desenvolvem-se perfeitamente fixados ao substrato, o seu crescimento ocorrendo essencialmente em área (forma achatada) ao longo dos limites dos talos e, marginalmente, em espessura [12, 13]. Assim, a medição bi-dimensional do seu crescimento é melhor efectuada em área [3].

1.2 Problema

A maioria dos trabalhos de liquenometria publicados, mais antigos mas também recentes, aplicam técnicas tradicionais de medição dos talos de líquenes [14–17]. Nestes trabalhos usam-se régua ou craveiras para efectuar medições o que introduz erros e aproximações grosseiras nos resultados obtidos [3, 9, 18]. Outro aspecto a ter em conta é que, com instrumentos de medição simples, não é possível medir directamente a área. Sendo necessário medir indicadores (e.g., eixo maior/menor) o que se pode tornar num problema de tempo e rigor (os líquenes não são perfeitamente cilíndricos/elípticos e com eixos perde-se informação). Estas técnicas clássicas de aquisição e processamento de dados apresentam alguns problemas, como sejam a ausência de replicabilidade, elevada variância dos dados colhidos por diferentes operadores, etc.

Estudos mais recentes de análise de cobertura de líquenes em rocha têm feito uso de câmaras digitais e software dedicado ao processamento de imagem para a obtenção dos seus resultados e análise de amostras [19]. No entanto, esse processo continua a ser um processo manual não automatizado, o que faz com que seja demorado e trabalhoso, especialmente quando o número de amostras é elevado. Para além disso, a sua dependência de decisões tomadas durante o procedimento aumenta a variância

das medidas feitas por mais do que um operador.

1.3 Proposta de solução e objectivos

Neste trabalho propõe-se desenvolver uma ferramenta de aquisição e processamento de imagem de forma a resolver os problemas associados à aquisição e processamento de dados para aplicação de liquenometria, nomeadamente a identificação, individualização e medição de líquenes. Propõe-se a utilização de técnicas de processamento de imagem digital, visão computacional e aprendizagem automática de forma a automatizar a segmentação e medição de líquenes em imagens, de modo a produzir metodologias e ferramentas para a captura e processamento de imagens para liquenometria. Pretende-se automatizar a recolha de informação, reduzindo o tempo de recolha de dados e facilitando o trabalho de campo (por exemplo selecção de talos a medir), pretende-se melhorar a exactidão na individualização e medição dos indivíduos em estudo (análise e processamento de dados recolhidos) e criar uma metodologia que possa ser replicada em qualquer local e por qualquer utilizador.

Devido à grande variedade de líquenes e superfícies que estes colonizam, um dos objectivos deste trabalho não é a criação de uma aplicação específica para processar os dados de uma determinada campanha (para uma espécie de líquen e superfície específica) mas sim desenvolver um sistema e metodologias que possam ser usadas no futuro, em campanhas diversas com variabilidade da espécie dos líquenes e tipo de superfícies. No entanto, o sistema foi desenvolvido para líquenes do tipo crustoso e foi testado apenas com líquenes desse tipo.

O sistema desenvolvido é composto por uma componente de hardware e de software. O hardware consiste numa câmara digital e miras especialmente definidas e avaliadas no âmbito desta dissertação para esta aplicação específica. A componente de software consiste numa série de algoritmos e metodologias de processamento das amostras adquiridas (imagens digitais) que produzem os seguintes resultados: segmentação das superfícies dos líquenes, percentagem de cobertura, contagem do número de indivíduos e tamanho de cada indivíduo (em mm²).

Os vários componentes foram avaliados de forma a compreender os benefícios do seu uso quando comparados com métodos não automatizados e perceber quais os erros associados. Nesta avaliação foram usados conjuntos de imagens referentes a 5 diferentes campanhas com variação na espécie dos líquenes e superfícies de colonização.

1.4 Estrutura do documento

Este documento foi organizado de forma a contextualizar o conteúdo do trabalho efectuado, o Capítulo 2 faz enquadramento com a área da biologia (explicando o propósito da liquenometria e

técnicas associadas), assim como, com a teoria das áreas de visão computacional e aprendizagem automática para problemas de classificação, o Capítulo 3 descreve a automatização do processamento das imagens, explicando detalhadamente cada componente do sistema (com alguns testes preliminares, que condicionaram a direcção do desenvolvimento), assim como, contextualizar do ponto de vista do utilizador. Foi feita também uma avaliação sistemática de cada área e componente do trabalho que culmina na avaliação e análise dos resultados finais, que é apresentado no Capítulo 4. No Capítulo 5 discute-se o trabalho efectuado e o documento termina com uma conclusão.

2

Trabalho relacionado

Conteúdo

2.1 Liquenometria	7
2.2 Teoria de visão computacional e classificação	11
2.3 Técnicas de visão computacional e classificação	13
2.4 Processamento de imagens aplicado à ecologia	20

Neste capítulo é feito o enquadramento da liquenometria e técnicas associadas, assim como é feita a introdução de certos conceitos e técnicas das áreas de visão computacional e aprendizagem automática para problemas de classificação. Acabando com a ilustração de alguns exemplos de estudos passados que abordam temas semelhantes.

2.1 Liquenometria

O estudo dos líquenes não é apenas pertinente em trabalhos acadêmicos na área da botânica. De facto, os líquenes têm sido utilizados em estudos diversos: a diversidade, estado e o tamanho de líquenes têm sido amplamente utilizados em estudos ecológicos relacionados com os efeitos da poluição atmosférica, alterações climáticas, degradação do solo e na estimativa da idade de depósitos geológicos ou estruturas arqueológicas, apenas para enumerar alguns [20–24].

A liquenometria é uma técnica de datação que é geralmente utilizada em superfícies rochosas recentemente expostas (<500 anos), em que a sua utilização combina conhecimentos de biologia (liquenologia) e de geologia, sendo utilizada na interpretação de formas de acumulação (depósitos) e de erosão (arribas). Esta técnica baseia-se na relação entre o tamanho dos líquenes e a sua idade. Se a taxa de crescimento de uma dada espécie for conhecida, por exemplo através de modelos de crescimento pré-determinados que relacionam a dimensão com a idade (Figura 2.1), o tempo decorrido desde a exposição da superfície colonizada por aquela espécie de líquenes pode então ser inferido.

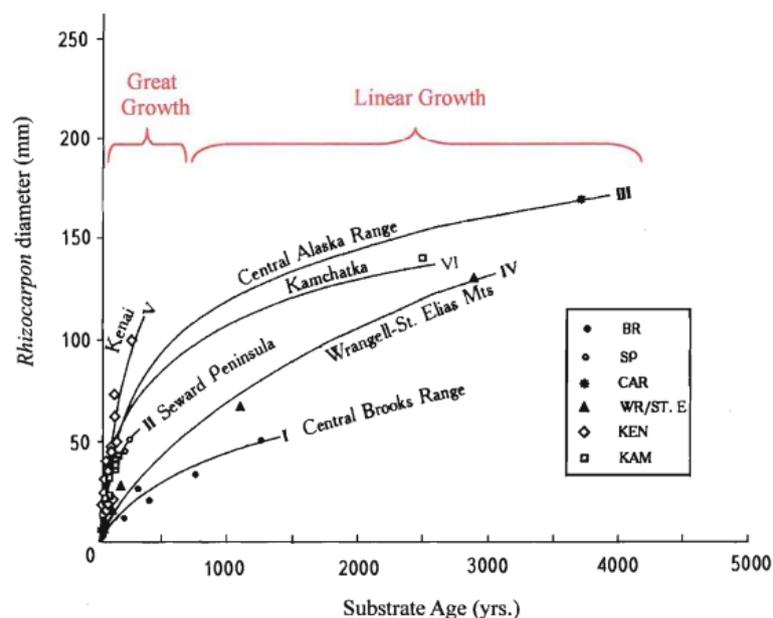


Figura 2.1: Comparação de curvas de datação liquenométricas do Alasca e arredores. [2]

Para se construir um modelo de crescimento de líquenes robusto é necessário efectuar muitas medições e avaliar a sua consistência no espaço, dado que a taxa de crescimento dos líquenes depende de vários factores, como por exemplo espécie e clima, apenas para enumerar alguns [17,25]. Assim, e com base nos diâmetros de líquenes em diversos pontos de idade conhecida e variada, obtêm-se curvas de crescimento representativas de uma região. A amostragem para aplicação em liquenometria requer a identificação das espécies de líquenes assim como a estimativa do tamanho do talo dos líquenes individuais (área ou diâmetro), para caracterizar a população na superfície de interesse.

Dependendo dos instrumentos de medição pode não ser possível medir directamente a área, sendo necessário medir outros indicadores. Diferentes autores utilizam diferentes parâmetros representativos de dimensão, como o eixo maior/menor, diâmetro do círculo circunscrito ou área dos talos para determinar a idade mínima em que a superfície colonizada por líquenes ficou exposta [2]. Contudo, tanto a identificação da espécie de líquenes como a medição do seu tamanho são demoradas, especialmente se for necessário aquisição de muitas amostras. Com dificuldades adicionais quando existe coalescência entre diferentes indivíduos, que complica e por vezes impede a sua individualização [3]. Na Figura 2.2(a) observa-se coalescência entre líquenes da mesma espécie (*O. durieui*), enquanto que na Figura 2.2(b) existe coalescência entre espécies diferentes (imagens pertencentes aos conjuntos usados na avaliação deste trabalho). É importante referir que a coalescência se torna um problema para a identificação de indivíduos quando ocorre entre indivíduos da mesma espécie ou de espécies muito semelhantes.

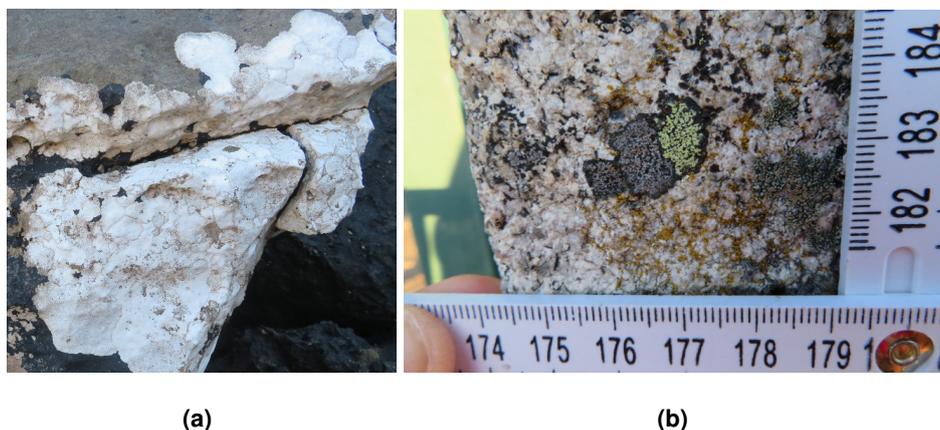


Figura 2.2: Exemplo de coalescência entre líquenes.

Dependendo do objectivo final do trabalho, superfícies específicas são seleccionadas para amostragem. Por exemplo, o tamanho dos líquenes que colonizam as superfícies de blocos em depósitos geológicos glaciares identificam fases distintas de deglaciações na Islândia [26]. De forma idêntica, o tamanho de líquenes foi também usado para diferenciar movimentos de massa de vertente (queda de

blocos de arribas) gerados por diferentes terremotos na Califórnia [27].

Recentemente foi efectuado um estudo de depósitos geológicos resultantes do transporte de blocos rochosos pela acção do mar por eventos marinhos extremos (tempestades e/ou tsunamis), na Costa Portuguesa, especificamente na região da Ericeira [3]. Os autores construíram um modelo de crescimento de uma espécie de líquenes (*O. durieui*) com base em várias superfícies de idade conhecida ao longo da costa oeste Portuguesa. Na Figura 2.3 pode-se observar uma das fotografias de líquenes capturadas numa superfície de idade conhecida num forte perto da Nazaré. A aplicação do modelo de crescimento desenvolvido à dimensão dos líquenes nos blocos rochosos transportados permitiu obter uma estimativa da idade mínima da formação dos depósitos geológicos [3]. Este trabalho vem no seguimento deste estudo geológico. A estimativa de idade destes depósitos é uma ferramenta essencial na reconstrução da cronologia de inundações marinhas extremas, que por sua vez é extremamente relevante para a avaliação de riscos costeiros [3].



Figura 2.3: Imagem de líquenes da espécie *O. durieui* capturada no forte S Miguel de Arcanjo, 25-01-2020, no âmbito de um estudo de análise de depósitos geológicos [3].

2.1.1 Técnicas de amostragem

A amostragem clássica utilizada em liquenometria baseava-se na selecção e medição, com recurso a craveira ou régua, do eixo maior ou do diâmetro do maior círculo circunscrito de talos de líquenes isolados (sem coalescência visível) [11]. A selecção dos líquenes era feita com base numa primeira aproximação visual quantitativa ao depósito ou estrutura em análise, sendo que os maiores (1, 5 ou 10 indivíduos) eram seleccionados para posterior medição. Esta selecção incorporava o princípio básico da liquenometria, de que quanto maior um indivíduo, mais velho, e maior a probabilidade de se medir os colonizadores iniciais da superfície, e assim, menor a probabilidade de subestimar a data de exposição da mesma.

Mais recentemente, e com o propósito de preservar e aumentar a quantidade de amostras colhidas no campo, começaram a ser utilizadas medições sobre imagens digitais. Com bases nestas imagens, foi assim possível posterior medição de diâmetro e áreas de talos individuais ou mesmo de áreas de cobertura de coalescências [3]. Outros trabalhos utilizaram técnicas mais complexas, com recurso a fotogrametria para comparação de dimensões de líquenes em superfícies 3D, colhidas em diferentes datas, com o objectivo específico de determinar taxas de crescimento directas [7]. Independentemente da maior ou menor complexidade na aquisição das imagens, o processamento de imagem (diferenciação entre líquen e rocha e delimitação dos líquenes) tem sido invariavelmente feito de forma manual ou semi-automática, a primeira através de vectorização sobre imagem e a última com recurso e ferramentas de selecção assistida em software de imagem.

A transformação de número de pixeis em dimensão requer a existência de elementos com dimensão conhecida, por exemplo incluindo escalas/réguas com representação milimétrica nas fotografias [3] (Figura 2.3), ou através da utilização de miras fotogramétricas de dimensão conhecida [7]. Com base nesta informação, torna-se assim possível georreferenciar as imagens e posteriormente extrair as dimensões/área ocupada pelos líquenes. Adicionalmente, obtém-se também uma medida de erro associada ao erro de georreferenciação das imagens.

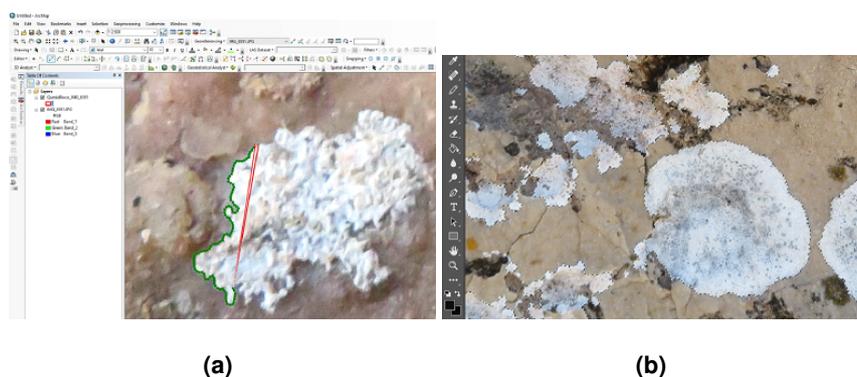


Figura 2.4: Exemplo de vectorização de um líquen com recurso à aplicação Esri® ArcMapTM e selecção de pixeis de líquenes com Adobe® Photoshop®. Imagens cedidas por M.A. Oliveira.

A vectorização de talos individuais é feita manualmente, tanto mais morosa quanto maior o nível de detalhe (ampliação da vectorização) ou a complexidade da forma dos talos (Figura 2.4(a)). A vectorização de talos individuais tem o potencial de produzir a melhor qualidade de informação, dado que cada talo se encontra representado por um polígono e dado a existência de um diverso conjunto de ferramentas geoespaciais em aplicações de sistemas de informação geográfica, que permitem medir automaticamente a área, ou mesmo delimitar o maior círculo circunscrito [3]. No entanto, é também a técnica mais morosa e, por ser manual, implica tomadas de decisão constantes relativamente à delimitação dos talos por parte do operador.

Alternativamente, a selecção assistida poderá ser mais rápida e com menor variação entre operador. Esta técnica baseia-se na detecção de limites por alterações de coloração em pixels, ou através da detecção de áreas com base numa pré-selecção de cores de interesse (Figura 2.4(b)) [28]. Apesar do auxílio prestado pelos algoritmos das aplicações de análise de imagem, a aplicação deste procedimento na selecção de líquenes requer constantes ajustes por parte do operador, tanto maiores quanto menos claros os limites dos líquenes ou quanto menor o contraste entre rocha e líquen. A utilização deste método é particularmente vantajosa na classificação de líquen/rocha, sobre imagens com menor ampliação e elevada coalescência, quando apenas se pretender calcular a área colonizada, não sendo por isso necessária posterior individualização de talos [3].

2.2 Teoria de visão computacional e classificação

A visão computacional utiliza algoritmia de modo a obter um entendimento de alto nível das imagens digitais [29]. A visão computacional foca-se na extracção, análise e compreensão automáticas de informações úteis de uma imagem. Os tópicos abordados nesta área englobam:

- Processamento de imagem
- Detecção de características - Pontos, zonas; detecção de contornos.
- Segmentação - Determinar grupos de pixels que pertencem ao mesmo objecto ou classe.
- Reconhecimento e detecção de objectos - Detectar objectos na imagem e reconhecer a que categoria ou classe pertencem.

Este trabalho foca-se especialmente na segmentação de imagem que é o processo de partição de uma imagem em múltiplos segmentos (conjunto de pixels). O objectivo é representar a imagem como algo mais significativo e mais fácil de analisar. Por outras palavras, a segmentação de imagem é o processo de atribuir uma classe a cada pixel de uma imagem, de modo a que os pixels com a mesma classe partilhem determinadas características [29]. No nosso caso, o fundo da imagem (correspondente à rocha) tem de ser isolado dos líquenes. A segmentação de imagem é uma tarefa extremamente simples, mas definitivamente não é uma tarefa trivial para computadores. O procedimento pode ser simplificado, por exemplo, convertendo a imagem para escala de cinzentos e aplicar um valor limiar. Outra alternativa considera pixels com gamas de cores iguais aos dos objectos presentes na imagem.

Quando o fundo é simples e não apresenta muita variabilidade, técnicas de detecção de contornos através do gradiente da imagem (e.g., direcção e intensidade dos pixels numa imagem em escala de cinzentos) podem ser uma maneira simples e viável de segmentar a imagem [29]. Métodos de análise

de texturas também se podem basear em metodologias de detecção de padrões através de histogramas de cor [29].

Aprendizagem automática e visão computacional são dois campos que se tornaram intimamente relacionados. A aprendizagem automática veio oferecer melhorias na área da visão computacional, contribuindo métodos eficazes para processamento de imagens, segmentação e identificação de objectos. As três categorias principais da aprendizagem automática são [30]:

- *supervised learning* - o computador recebe um conjunto de entradas e saídas desejadas e tem como objectivo encontrar o mapa entre elas, que aplica a novas entradas não conhecidas.
- *unsupervised learning* - o computador recebe um conjunto de entradas mas nenhuma saída desejada. O objectivo é encontrar a estrutura dos dados (distribuição de probabilidades, grupos), que aplica a novas entradas não conhecidas.
- *reinforced learning* - visa aprender o comportamento dos agentes de software ou robôs com base no *feedback* do ambiente.

A aprendizagem automática aborda vários problemas sendo dois dos mais comuns problemas de regressão e problemas de classificação. De um modo simples quando o modelo produz um escalar diz-se ser um problema de regressão. A classificação refere-se a um problema de modelação preditiva em que se prevê uma etiqueta de classe para um dado exemplo de dados de entrada. A classificação binária refere-se à previsão de uma de duas classes e a classificação multiclasse envolve a previsão de mais de duas classes. Este trabalho foca-se apenas na classificação binária para as classes de líquen e rocha/outro.

Pretende-se com isto atribuir a cada pixel uma de duas classes:

- Classe líquen - classe atribuída a pixels pertencentes a superfícies de líquenes.
- Classe fundo - classe atribuída a pixels não pertencentes a superfícies de líquenes.

Alguns algoritmos populares que podem ser utilizados para a classificação (binária) são [31]:

- *Logistic Regression*
- *k-Nearest Neighbors*
- *Decision Trees*
- *Support vector machine (SVM)*
- *Naive Bayes*

As abordagens comuns que usam *deep learning* para reconhecimento de objectos são baseadas em *Convolutional neural network* (CNN). Uma rede neuronal convolucional é um tipo de rede artificial com várias camadas entre a entrada e a saída, em que certas camadas efectuam operações de convolução [30]. Uma rede neuronal artificial é um sistema de computação inspirado na rede neuronal biológica do cérebro. Um exemplo é a rede AlexNet CNN, projectada para classificar imagens do banco de dados ImageNet [32]. Com o sucesso da AlexNet, descobriu-se que técnicas semelhantes podem ser generalizadas para outros conjuntos de dados e tarefas de visão computacional, como detecção de objectos [33] e segmentação de imagem [34].

No entanto, para certos problemas relativamente simples (ou com poucos dados), os métodos tradicionais são mais eficientes e apresentam bons resultados, enquanto que soluções de *deep learning* requerem mais capacidade de processamento e necessitam de um conjunto de dados grande e devidamente caracterizado (no caso do *supervised learning*) [35].

Este trabalho foca-se na utilização de dois tipos de algoritmos da área de *supervised machine learning* o *Random Forests* (algoritmo baseado em *decision trees*) e SVM para classificação binária, que serão descritos na próxima secção.

2.3 Técnicas de visão computacional e classificação

Nesta secção são descritas as várias técnicas e métricas de visão computacional e aprendizagem automática utilizadas neste trabalho.

2.3.1 Classificação manual com GrabCut

A maioria das técnicas de segmentação faz uso da informação sobre as regiões e fronteiras das regiões contidas na imagem a fim de realizar a segmentação. O GrabCut [36] é uma técnica de segmentação que utiliza ambas para realizar a segmentação.

Para realizar a segmentação é construído um grafo, onde os nós do grafo representam pixels na imagem. Para além disso, são também criados dois nós especiais, os nós *Sink* e *Source* (Figura 2.5) marcados pelo utilizador na imagem como *background* e *foreground* correspondentemente. Cada nó de pixel no grafo é ligado ao nó *Source* e ao nó *Sink*. O nó *Source* representa o primeiro plano da imagem (os pixels que queremos extrair, *foreground*), e o nó *Sink* o segundo plano (*background*). A fim de segmentar a imagem, os nós *Source* e *Sink* têm de ser separados.

Uma função de energia/custo é incorporada no grafo como pesos entre nós de pixel e pesos entre pixel e nós *Source* ou *Sink*. Os pesos entre nós de pixels são determinados por estarem ou não localizados numa região fronteira na imagem. Assim, uma forte indicação de uma fronteira entre dois pixels (uma grande diferença na cor dos pixels) resulta num peso muito pequeno entre dois nós de

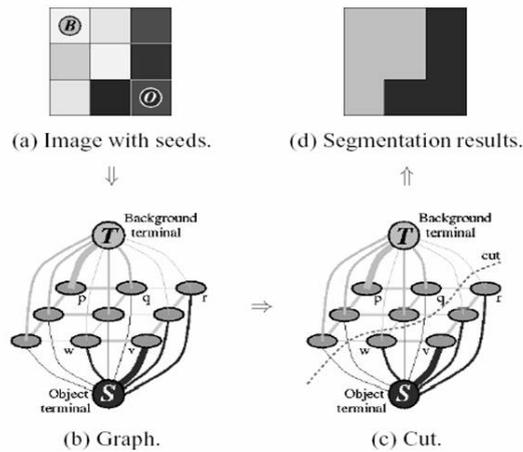


Figura 2.5: Exemplo da segmentação 2D de uma imagem 3x3 [4].

pixel (analogamente, o peso entre dois nós de pixels com cor semelhante será elevado). A informação da região determina o peso entre os nós de pixels e os nós de *Source* e *Sink*. Estes pesos são calculados determinando a probabilidade do nó de pixel fazer parte da região de fundo ou de primeiro plano (essa probabilidade é dada por um *Gaussian mixture model* (GMM) modelado com os inputs iniciais de *background* e *foreground* marcados pelo utilizador).

Para que as regiões de primeiro plano e segundo plano sejam criadas, alguns pixels na imagem precisam de ser marcados pelo utilizador antes da segmentação como primeiro plano ou segundo plano. Quaisquer pixels que são marcados durante esta fase são definidos como restrições. Isto significa que, durante o processo de segmentação, a classificação dos pixels como primeiro ou segundo plano identificados pelo utilizador não se modifica.

Um algoritmo *Min-cut/Max-Flow* é utilizado para segmentar o grafo. Este algoritmo determina o corte de custo mínimo, determinado pela soma de todos os pesos das ligações que são cortadas (minimização da função de energia/custo), que irá separar os nós de *Source* e *Sink*. Uma vez separados os nós *Source* e *Sink*, todos os nós de pixel ligados ao nó *Source* tornam-se parte do primeiro plano, e os restantes tornam-se parte do segundo plano.

Resumindo o GrabCut é um método de segmentação de imagem baseado em cortes de grafos. Começando com uma caixa de delimitação especificada pelo utilizador em torno do objecto a ser segmentado, o algoritmo estima a distribuição da cor do objecto alvo e a do fundo, utilizando um modelo de mistura Gaussiano (GMM) [36]. Isto é utilizado para construir um grafo sobre as classes de pixels, com uma função de energia que prefere regiões ligadas com a mesma classe, e executando uma optimização baseada em cortes de grafos para inferir os seus valores [4].

As estimativas podem ser ainda corrigidas pelo utilizador, apontando regiões mal classificadas e executando a optimização, que executa de forma iterativa os seguintes passos:

- Estimar a distribuição da cor do primeiro plano e do fundo através de um Modelo de Mistura Gaussiano (GMM).
- Construção de um grafo sobre as classes dos pixels (ou seja, primeiro plano vs. fundo).
- Aplicação de uma otimização do corte do grafo (algoritmo *Min-cut/Max-Flow*) para chegar à segmentação final.

2.3.2 Classificação automática com SVM e Random Forests

Na aprendizagem automática, SVM são modelos de aprendizagem supervisionados com algoritmos de aprendizagem associados que analisam dados para classificação e análise de regressão [30]. Dado um conjunto de dados de treino, constituído por imagens já classificadas, um algoritmo de treino SVM constrói um modelo que atribui novos dados a uma ou outra classe, tornando-o um classificador linear binário não-probabilístico. Os SVM mapeiam dados de treino para pontos no espaço e procuram maximizar a largura do intervalo entre os pontos das duas classes (no caso da classificação binária só existem duas classes). Novos dados são então mapeados para esse mesmo espaço e prevê-se que pertençam a uma das duas classes com base no lado da fronteira em que se situam. Para além de realizarem uma classificação linear, os SVM podem realizar eficientemente uma classificação não linear utilizando o que se chama o *kernel trick*, mapeando implicitamente as suas entradas em espaços de características de alta dimensão.

Mais formalmente, uma SVM constrói um hiperplano ou conjunto de hiperplanos num espaço de alta dimensão, que pode ser utilizado para classificação, regressão, ou outras tarefas como a detecção de *outliers* [37]. Intuitivamente, no caso da classificação, uma boa separação é alcançada pelo hiperplano que tem a maior distância para o ponto de dados de treino mais próximo de qualquer classe, uma vez que em geral quanto maior for a margem, menor será o erro de generalização do classificador [30].

Enquanto que o problema original pode ser definido num espaço dimensionalmente finito, é frequente os conjuntos a discriminar não serem linearmente separáveis nesse espaço. Por este motivo, foi proposto que o espaço original seja mapeado para um espaço muito mais elevado, presumivelmente facilitando a separação nesse espaço [38]. Para manter a carga computacional razoável, os mapeamentos utilizados pelos esquemas SVM são concebidos para assegurar que os produtos escalares dos pares de vectores dos dados de entrada possam ser calculados facilmente em termos das variáveis no espaço original, definindo-as em termos de uma função de *kernel* $k(x, y)$ seleccionado para se adequar ao problema [39]. Um classificador que utilize SVM como algoritmo de suporte é comumente denominado de *Support vector classifier* (SVC).

O *Random Forests* é um tipo particular de algoritmo de aprendizagem baseado em árvores de

decisão. Na aprendizagem com árvores de decisão utilizam-se árvores de decisão como modelos de previsão para ir desde as observações sobre um item (representado nos ramos) até conclusões sobre o valor-alvo do item (representado nas folhas). Os modelos de árvores onde a variável alvo pode tomar um conjunto discreto de valores, são chamados árvores de classificação; nestas estruturas de árvores, as folhas representam as etiquetas da classe e os ramos representam conjunções de características que levam a essas etiquetas de classe.

Algumas técnicas, muitas vezes chamadas *ensemble methods*, constroem mais do que uma árvore de decisão. Árvores de decisão *bootstrap aggregated (bagged)* constroem múltiplas árvores de decisão através da reamostragem repetida de dados de treino com substituição, e votação das árvores para uma previsão consensual [40]. Um classificador *random forest* é um tipo específico de *bootstrap aggregating*. *Random forests* correspondem a um método *ensemble* de aprendizagem para classificação, regressão e outras tarefas, que funciona através da construção de uma multiplicidade de árvores de decisão no período de treino. Para tarefas de classificação, o resultado do *random forests* é a classe seleccionada pela maioria das árvores [30].

2.3.3 Validação cruzada para estimar hiperparâmetros

A afinação dos hiperparâmetros dos classificadores depende muito dos conjuntos de dados, pelo que um bom método para determinar os parâmetros óptimos é testar muitas combinações diferentes para avaliar o desempenho de cada modelo. No entanto, avaliar cada modelo apenas no conjunto de treino pode levar a um dos problemas mais fundamentais na aprendizagem de máquinas: o *overfitting* [30]. Se optimizarmos o modelo para os dados de treino, então o nosso modelo terá uma pontuação muito boa no conjunto de treino, mas não será capaz de generalizar a novos dados, como num conjunto de teste. O *overfitting* resulta da criação de um modelo que conhece muito bem o conjunto de treino, mas não pode ser aplicado a novos dados. Por conseguinte, o procedimento padrão para a optimização de hiperparâmetros de modo a evitar o *overfitting* é a validação cruzada ou *Cross validation (CV)*.

A técnica de validação cruzada (CV) é melhor explicada pelo exemplo utilizando o método mais comum, *K-Fold CV*. Quando se aborda um problema de aprendizagem automática, certifica-se que se dividem os dados em conjuntos de treino e teste. No *K-Fold CV*, divide-se ainda mais o conjunto de treino em K número de subconjuntos, chamados *folds*. A título de exemplo, ao considerar encaixar um modelo com $K = 5$, na primeira iteração treinam-se os primeiros quatro *folds* e avalia-se no quinto *fold*. Na segunda iteração treinam-se o primeiro, segundo, terceiro e quinto *folds* e avalia-se no quarto *fold*. Repete-se este procedimento mais 3 vezes, avaliando cada vez num *fold* diferente. No final do treino, calcula-se a média do desempenho em cada uma dos *folds* para se obter a métrica final de validação do modelo como se pode observar no exemplo da Figura 2.6.

Para afinação de hiperparâmetros, realizam-se muitas iterações de todo o processo de *K-Fold CV*,

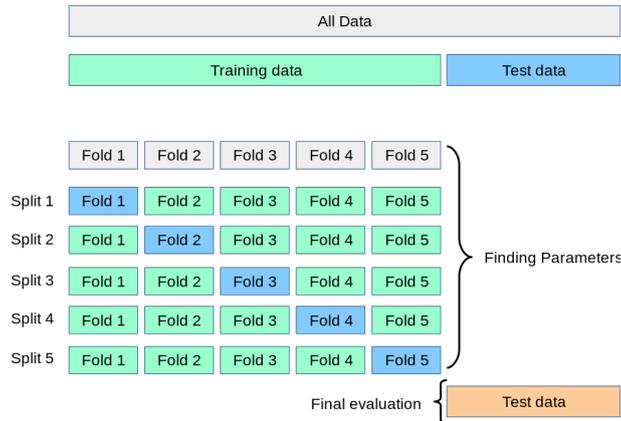


Figura 2.6: Diagrama de validação cruzada [5].

cada vez utilizando configurações de modelos diferentes. Assim, todos os modelo são comparados, o melhor é seleccionado, treina-se com conjunto de treino completo e finalmente, avalia-se com o conjunto de teste. Apesar de ser demorado, este processo é necessário sempre que se quer avaliar um conjunto diferente de hiperparâmetros, tendo de se dividir os dados de treino em K -folds e treinar e avaliar. Como exemplo, 10 conjuntos de hiperparâmetros e um CV com 5 folds, representa 50 ciclos de treino.

Alguns problemas de classificação podem apresentar um grande desequilíbrio na distribuição das classes-alvo: por exemplo no caso binário, pode haver várias vezes mais amostras de uma classe do que amostras da outra classe. Nesses casos, recomenda-se a utilização de amostragem estratificada, tal como implementado na função *StratifiedKFold* do *scikit-learn* (biblioteca de aprendizagem automática em *python*), para assegurar que as frequências relativas das classes sejam aproximadamente preservadas em cada cadeia e *fold* de validação.

StratifiedKFold (exemplificado na fig. 2.7) é uma variação do k -folds que retorna os folds estratificados: cada conjunto contém aproximadamente a mesma percentagem de amostras de cada classe alvo do conjunto completo .

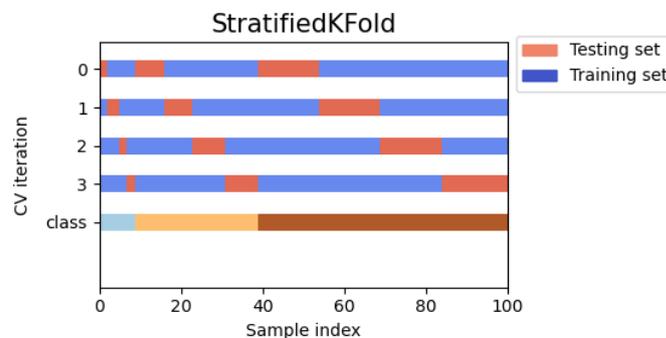


Figura 2.7: *Stratified k-fold* exemplo com 4 folds e 3 classes [5].

Os melhores hiperparâmetros (do `RandomForestClassifier` e `SVC`) foram determinados através da técnica de pesquisa em grade implementada pela função `Gridsearch` do `scikit-learn` que faz uma pesquisa exaustiva sobre valores de parâmetros especificados para um estimador. No caso do número de *folds* especificado for um número inteiro (neste caso utilizaram-se 5 *folds*) e o estimador for um classificador, a função utiliza o `StratifiedKFold`.

2.3.4 Simple Linear Iterative Clustering (SLIC)

O *Simple Linear Iterative Clustering* (SLIC) [41] cria super-píxeis com base no agrupamento *k-means*. Os super-píxeis são pequenas regiões de píxeis da imagem que partilham propriedades semelhantes (cor). Os super-píxeis simplificam as imagens com um grande número de píxeis tornando-as mais fáceis de serem tratadas em muitos domínios (visão por computador, reconhecimento de padrões e aprendizagem automática). Este algoritmo gera super-píxeis agrupando os píxeis com base na sua semelhança de cor e proximidade no plano da imagem.

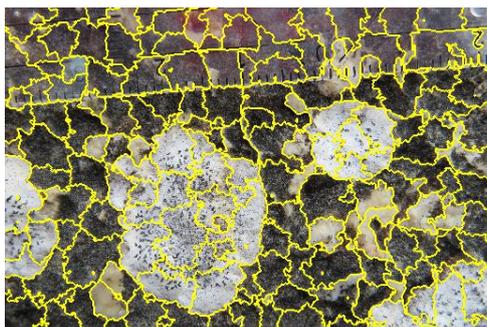


Figura 2.8: Exemplo das regiões dos segmentos SLIC para os parâmetros: *n_segments*=250, *compactness*=10 e *sigma*=1.

A função SLIC da biblioteca de *python*, *skimage*, tem os seguintes parâmetros:

- *n_segments* : *integer*, opcional. Número (aproximado) de segmentos SLIC criados para a imagem de entrada.
- *compactness* : *float*, opcional. Equilibra a proximidade de cores e a proximidade do espaço. Valores superiores dão mais peso à proximidade do espaço, tornando as formas super-píxel mais quadradas. Este parâmetro depende fortemente do contraste da imagem e das formas dos objectos na imagem. Recomenda-se a exploração de possíveis valores numa escala logarítmica, por exemplo, 0,01, 0,1, 1, 10, 100, antes de refinar em torno de um valor escolhido com melhor desempenho.
- *sigma* : *float* ou (3,) vector de *floats*, opcional. Largura do *kernel* de suavização gaussiano para o pré-processamento de cada dimensão da imagem. O mesmo *sigma* é aplicado a cada dimensão

no caso de um valor escalar. Zero significa que não há suavização.

- *threshold* (este parâmetro não pertence á função SLIC da biblioteca de *python*, *skimage*, mas foi adicionado por ser necessário transformar as imagens *label* (pixeis a preto e branco) em segmentos SLIC cada um classificado como pertencente à classe líquen ou rocha, já que existe a possibilidade de um segmento SLIC comportar pixeis de ambas as classes): quando considerar uma região SLIC como pertencente à classe líquen ou não, quando comparada com o *label* do grabcut (e.g. para um *threshold* de 0.5 se mais de metade da região SLIC pertencer a uma zona de líquen na imagem de *label* correspondente então todo o segmento será considerado como líquen)

O uso do SLIC permite treinar os classificadores, decidindo-se dividir as imagens em regiões dadas pelo SLIC. Testes preliminares efectuados no decorrer deste trabalho indicam que esta abordagem providencia uma boa delimitação entre líquenes e rocha, ou seja, as regiões fronteira dos segmentos SLIC coincidem, em grande parte, com as fronteiras entre líquenes e rocha (Figura 2.8). Desta forma reduz-se de uma forma drástica o número de características (número de características fica igual ao número de segmentos SLIC em vez do número de pixeis).

2.3.5 Matthews correlation coefficient (MCC)

O *Matthews correlation coefficient* (MCC) é utilizado na aprendizagem automática como medida da qualidade das classificações no caso binário (duas classes), introduzido pelo bioquímico Brian W. Matthews em 1975 [42].

O coeficiente tem em conta os verdadeiros e falsos positivos e negativos e é geralmente considerado como uma medida que pode ser utilizada mesmo que as classes sejam desequilibradas [43]. Neste trabalho consideram-se como verdadeiros positivos os pixeis de líquen correctamente classificados, verdadeiros negativos os pixeis não líquen (fundo) correctamente classificados, falsos positivos os pixeis que deveriam ter sido classificados como fundo mas foram classificados como líquen e falsos negativos os pixeis que deveriam ter sido classificados como líquen mas foram classificados como fundo.

O MCC é essencialmente um coeficiente de correlação entre as classificações binárias observadas e previstas; retorna um valor entre -1 e +1. Um coeficiente de +1 representa uma previsão perfeita, 0 não melhor que a previsão aleatória e -1 indica um desacordo total entre a previsão e a observação.

Embora não exista uma forma perfeita de descrever a matriz de confusão (tabela) de verdadeiros e falsos positivos e negativos por um único número, o MCC é geralmente considerado como sendo uma das melhores medidas deste tipo [44]. Outras medidas, tais como a proporção de previsões correctas (precisão), não são úteis quando as duas classes possuem dimensões muito diferentes.

O MCC pode ser calculado directamente a partir da matriz de confusão usando a seguinte equação:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.1)$$

Na Equação (2.1), TP (*true positive*) é o número de positivos verdadeiros, TN (*true negative*) o número de negativos verdadeiros, FP (*false positive*) o número de falsos positivos e FN (*false negative*) o número de falsos negativos. Se qualquer das quatro somas no denominador for zero, o denominador pode ser arbitrariamente fixado em um; isto resulta num MCC de zero.

2.4 Processamento de imagens aplicado à ecologia

Várias técnicas de visão computacional têm sido aplicadas no contexto da análise de vegetação. Estas técnicas envolvem, em geral, segmentação e classificação de imagens com o interesse de medir certas características geométricas como por exemplo área das folhas, altura ou volume (para estimar crescimento), assim como identificar espécies ou possíveis plantas afectadas por doenças [45, 46].

O procedimento frequentemente utilizado em trabalhos de processamento de imagem aplicados à caracterização da vegetação [45, 47] é segmentar as imagens através da definição de limiares de um certo tom de cor ou de intensidade no caso da escala de cinzentos. Por exemplo os histogramas de cor das imagens podem ser utilizados para escolher a tonalidade que se pretende segmentar [45]. Seguindo como exemplo um estudo de análise de alfices num contexto agrícola [47], uns dos primeiros passos a seguir é a segmentação da planta. A extracção do contorno é necessária para distinguir um objecto do fundo. Uma das metodologias mais simples baseia-se na transformação da imagem *Red, green, blue* (RGB) para escala de cinzentos e na definição de um limiar que defina a partir de que valor os pixels pertencem ao objecto ou ao fundo. Assim, produz-se uma imagem binária da imagem original. O valor do limiar pode ser determinado estatisticamente, pelo uso de uma função discriminante, ou através de um algoritmo apropriado [48].

Métodos alternativos baseiam-se numa análise pela cor de cada pixel transformando a imagem do modelo de cor RGB para o espaço de cores *Hue, saturation, value* (HSV) [19, 49]. O substrato e cada espécie de planta podem ser definidos através de grupos de pixels pelas suas coordenadas HSV de acordo com grupos de cores específicos que incluíam várias classes de cores, identificadas pela matiz, saturação e intensidade (modelo de cores HSV). Em certos casos, é realizada filtragem morfológica (filtros de gauss, mediana) para suavizar os contornos dos objectos e eliminar pequenos objectos e ruído ou análise estatística para remover possíveis *outliers* [50, 51]. Depois de segmentada a região de interesse, podem ser efectuadas as medições necessárias.

Alguns exemplos da utilização de métodos de aprendizagem automática e processamento de imagens no contexto de visão computacional relacionado com a análise de vegetação, incluem:

- Identificar o *threshold* num problema de segmentação binária para remover *background* em imagens de alfaces em escala de cinzentos, utilizando o método de Otsu, de forma a automatizar as medições da área das alfaces [49] (na área da visão computacional e processamento de imagens, o método de Otsu é utilizado para seleccionar automaticamente o limiar de uma imagem em escala de cinzentos [52]). Na forma mais simples, o algoritmo retorna um único *threshold* de intensidade que separa os pixels em duas classes, primeiro plano e segundo plano.
- Identificação de um índice de vegetação para líquenes com base em medições hiperespectrais (no espectro visível ao infra-vermelho médio) usando amostras como conjuntos de dados de treino e validação para encontrar os valores óptimos, minimizando o *Root mean squared error* (RMSE). A mistura espectral de líquenes e rochas pode dificultar o diagnóstico de materiais de interesse, levando assim a interpretações erradas e falsos positivos se o mapeamento for feito com base em metodologias de correspondência espectral. Portanto, a capacidade de distinguir a cobertura do líquen da rocha e decompor um pixel misturado em uma colecção de espectros de pura reflectância, pode melhorar a aplicabilidade de métodos hiper-espectrais para a exploração mineral. [53]

No entanto até à data não existe uma solução que resolva os problemas de amostragem para aplicação de liquenometria.

3

Automatização do processamento de imagens

Conteúdo

3.1	Captura de fotografias	24
3.2	Correcção de fotografias	25
3.3	Processo de classificação manual	26
3.4	Classificação automática	27
3.5	Calculo das áreas dos líquenes	34

Os objectivos deste projecto prendem-se com a criação de uma ferramenta de medição automática de dimensões de líquenes de forma a contribuir para uma melhor aquisição de dados para datação com liquenometria em estudos geológicos e/ou arqueológicos. Os objectivos podem ser particularizados da seguinte forma:

- Detecção de líquenes.
- Medição da área dos líquenes.

A Figura 3.1 apresenta o fluxo do utilizador.

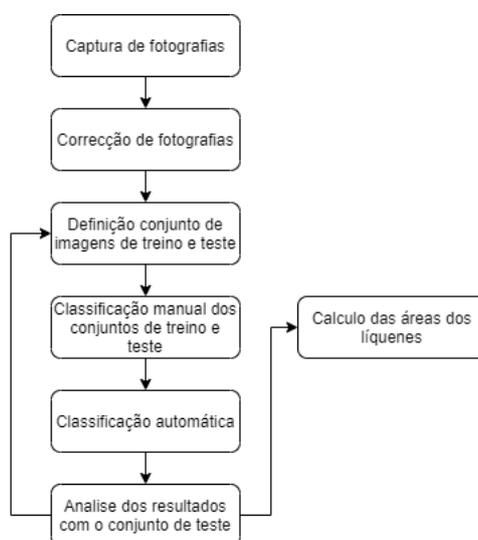


Figura 3.1: Fluxo do utilizador.

O fluxo do utilizador comporta os seguintes passos:

- Captura de fotografias - A ferramenta será aplicada a conjuntos de dados previamente amostrados e a novos conjuntos de dados adquiridos no decorrer deste projecto. Cabe ao utilizador escolher qual o conjunto de dados que se pretende analisar com o programa.
- Correcção de fotografias - Dado o conjunto de dados escolhido, é efectuada a correcção de cada fotografia do conjunto de dados de modo a corrigir erros de perspectiva.
- Definição do conjunto de imagens de treino e teste - Definição de dois sub-conjuntos (treino e teste) de imagens pertencentes ao conjunto de dados escolhido. As imagens de treino servem para treinar o classificador a identificar as regiões de interesse (líquenes) e as imagens de teste servem para avaliar e aferir o desempenho do sistema.
- Classificação manual dos conjuntos de treino e teste - As regiões de interesse (líquenes) das imagens de treino e teste são explicitamente identificadas, classificando manualmente estas imagens

com o auxílio de uma ferramenta de extração de *background*. Disto resultam imagens binárias com as regiões da classe líquen a branco e o resto a preto. Estas imagens, conjuntamente com as imagens de treino e teste correspondentes, é que permitem efectuar a aprendizagem do programa.

- Classificação automática - O programa irá classificar automaticamente as regiões correspondentes a líquenes das imagens dos conjuntos de dados.
- Análise dos resultados com o conjunto de teste - O sistema retorna as imagens segmentadas e o desempenho das medições efectuadas sobre o conjunto de imagens de teste. Dependendo do desempenho, pode necessário voltar a executar o programa com mais imagens de treino.
- Cálculo das áreas dos líquenes - Para cada segmentação escolhida, efectua-se a medição da área de cada talo.

Nas próximas secções descreve-se cada elemento presente na Figura 3.1 em detalhe.

3.1 Captura de fotografias

Um dos problemas associados à amostragem com recurso a fotografia, particularmente relevante quando o objectivo é extrair informação espacial (tal como a área ocupada por um talo de líquen), é a deformação da imagem. Esta ocorre quer devido à orientação oblíqua da fotografia relativamente à superfície, quer devido à deformação da lente.

Assim, na aquisição de novas imagens no terreno, propõe-se o uso de 4 miras azuis dispostas sobre os vértices de uma região quadrada/rectangular que inclua a zona de interesse a ser fotografada. Isto irá permitir a aplicação de correcções e transformações de modo a compensar erros de perspectiva e atribuir, uma escala.

As distâncias, nomeadamente os valores da largura e altura da região quadrada delimitada pelas miras, têm de ser registadas durante a amostragem para mais tarde se introduzir no programa para que seja possível remover a deformação da imagem.

Este processo fornece uma melhor precisão de medição dos líquenes do que a utilização de escalas/réguas. Na Figura 3.2 pode-se observar aquisição sem instrumentação de escala, com régua/esquadro e com o sistema de miras. Na primeira figura não é possível obter escala, na segunda é possível porque tem régua mas o processo é manual, e na terceira figura, com o sistema de miras, este processo é possível de ser automatizado.

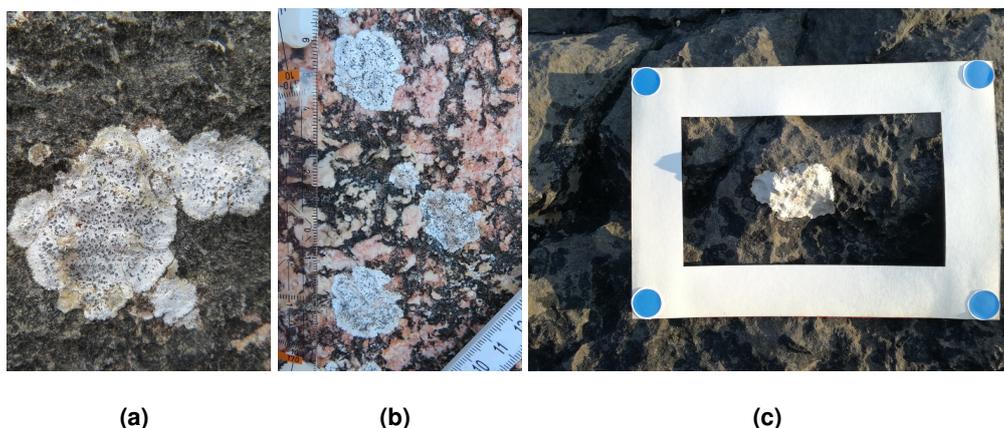


Figura 3.2: Exemplos de diferentes tipos de aquisição.

3.2 Correção de fotografias

Nesta secção é explicado de que forma as miras desenvolvidas (Figura 3.2(c)), são usadas para corrigir os erros de perspectiva das imagens, assim como obter a escala.

Numa primeira iteração do sistema de miras, fez-se a detecção das miras (da marca Leica©) através de descritores *Scale-invariant feature transform* (SIFT) [54] e do uso do *Random sample consensus* (RANSAC). O SIFT gera descritores que representam pontos notáveis numa imagem permitindo corresponder os descritores SIFT da imagem de pesquisa (imagem só com a mira a detectar) aos descritores SIFT da imagem distorcida (a correspondência será feita com uma das 4 miras da imagem). Esta correspondência é efectuada com o RANSAC sabendo que uma correspondência só pode existir entre pontos pertencentes ao mesmo plano (homografia). Ora como estas miras eram todas iguais, sempre que se fazia uma correspondência entre a imagem de pesquisa e uma mira no local era necessário o programa tapar essa mira da imagem. Caso contrário, a correspondência seria sempre feita entre os mesmos descritores SIFT, pertencentes à mesma mira. A Figura 3.3 mostra o processo de detectar 2 miras com a ocultação das miras já identificadas.

Este método, apesar de produzir alguns resultados promissores em experiências controladas, não é tão robusto para efectuar a detecção das miras para imagens com fundos rochosos ricos em características SIFT. Mais importante, quando a deformação da perspectiva da imagem é mais intensa, este sistema deixa de conseguir detectar todas as 4 miras e começa a corresponder as miras a pontos errados.

De modo a melhorar a robustez, decidiu-se partir para uma solução mais simples através de segmentação por cor, usando 4 miras circulares com a mesma cor azul. Esta cor foi escolhida de forma a proporcionar um bom contraste e por ser uma cor pouco comum, tanto em formações rochosas como em líquenes.

De forma a determinar os parâmetros necessários à segmentação das miras por cor, as imagem

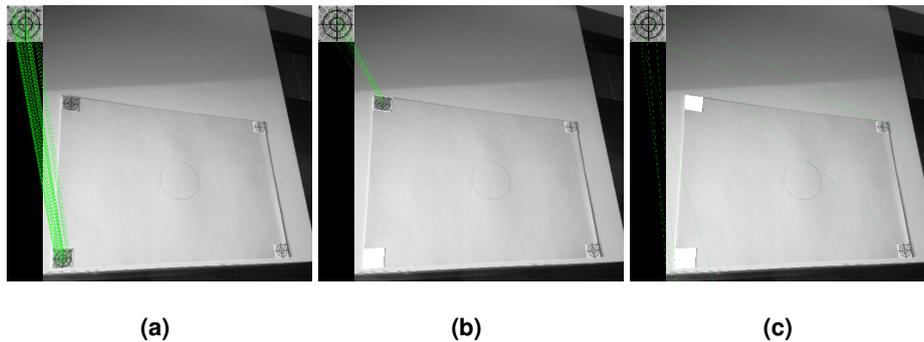


Figura 3.3: Exemplificação de erro na detecção da terceira mira, do sistema com SIFT e miras Leica®.

fotografadas com o sistema de miras foram, numa primeira instância, convertidas de RGB para HSV. Isto torna mais fácil a definição dos limites inferiores e superiores dos canais de cor de forma a segmentar as miras. Os intervalos H, S e V definidos experimentalmente foram: H[95-105], S[85-255] e V[170-245].

O sistema detecta os centros de massa das regiões segmentadas, correspondentes às miras. Com as coordenadas desses 4 pontos e sabendo a geometria real das miras (rectângulo com 27.2cm de comprimento e 18.5cm de altura, por exemplo) é calculada a matriz de transformação geométrica. Com esta matriz transforma-se a perspectiva da imagem de modo a eliminar os erros da perspectiva original e melhorar a qualidade das medições.

Observa-se na Figura 3.4 que a deformação do alvo circular no centro da região das miras é corrigida.

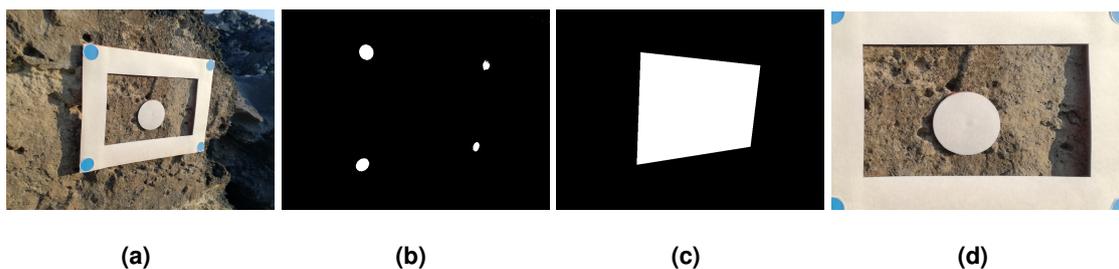


Figura 3.4: Exemplificação do funcionamento do sistema com segmentação de cor e miras azuis.

3.3 Processo de classificação manual

A classificação manual gera outputs binários correspondentes às imagens de treino e teste, em que as regiões pertencentes à classe de líquen são identificadas a branco e as restantes a preto, permitindo ao utilizador "ensinar" ao programa quais as características de interesse das imagens. Estas imagens

binárias, conjuntamente com as imagens de treino e teste correspondentes, servem como referencia para efectuar a aprendizagem do programa.

O processo de classificação manual das imagens é feito através do algoritmo GrabCut.

Do ponto de vista do utilizador, o algoritmo GrabCut aceita uma imagem de entrada onde: (1) o utilizador, usando o rato, marca diversas áreas correspondentes aos líquenes; (2) usando o rato, o utilizador marca diversas áreas correspondentes ao fundo; (3) o sistema actualiza o resultado binário e se o utilizador desejar poderá marcar novas áreas. Por vezes, a segmentação efectuada pelo algoritmo com base na delimitação do utilizador, está longe de ser a ideal. Nesses casos, o utilizador precisa de fazer retoques finos que irão modificar a classificação, seleccionando nas imagens onde existem alguns resultados defeituosos e marcando-os devidamente.

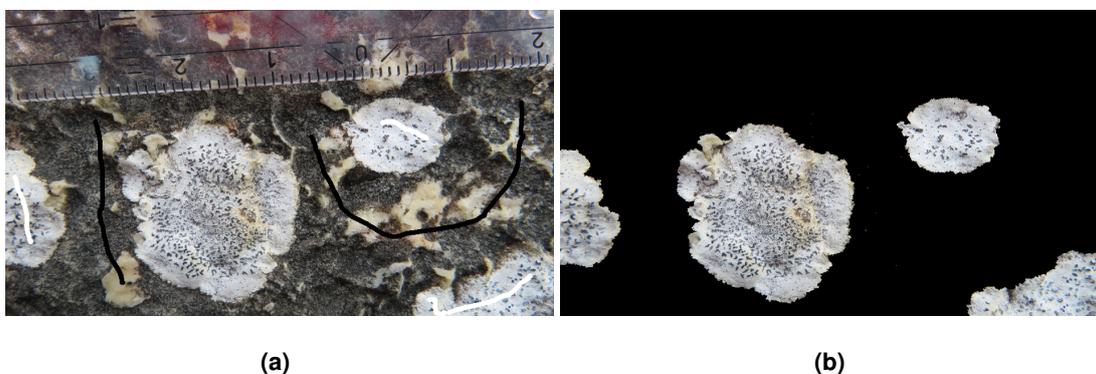


Figura 3.5: Exemplo da selecção de primeiro plano (traços brancos) e de segundo plano (traços pretos) e classificação final com recurso ao GrabCut.

Como se pode observar na Figura 3.5, depois de alguns retoques finais, identificados pelos traços brancos (denotando primeiro plano) e pretos (denotando o fundo) obtém-se um bom resultado de segmentação que, neste caso, separa líquen de rocha.

3.4 Classificação automática

Nesta secção são explicadas todas as etapas e funcionamento das componentes que efectuem a classificação automática.

O programa de classificação automática de líquenes recebe como dados de entrada a localização da directoria onde se encontram as imagens (conjunto de dados). O utilizador tem que definir o número de imagens que serão utilizadas para treino e igualmente para teste (escolhidas aleatoriamente). Estas fotografias serão classificadas manualmente, como descrito na Seção 3.3. Estes conjuntos de dados permitem treinar os classificadores com dados devidamente classificados e permitem obter uma

referência para avaliação do desempenho da classificação automática do programa. O fluxo de funcionamento do programa está ilustrado na Figura 3.6.

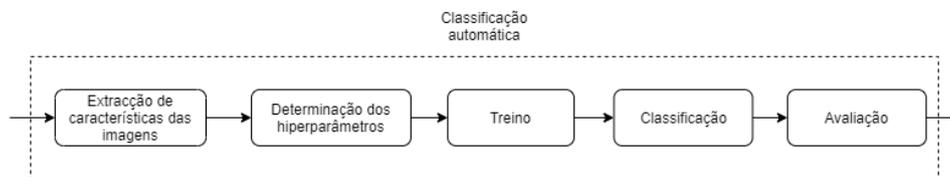


Figura 3.6: Fluxo da classificação automática.

A componente da classificação automática recebe as imagens de treino e teste, assim como as imagens binárias correspondentes, classificadas manualmente com o GrabCut. Na saída são retornadas as segmentações das restantes imagens do conjunto de dados (imagens que não pertencem aos conjuntos de treino e teste).

3.4.1 Extração de características das imagens

O programa utiliza como características os histogramas de frequência relativa correspondentes aos segmentos SLIC das imagens. Em alternativa poder-se-ia usar cada pixel como característica para treino e classificação. No entanto, o uso dos segmentos SLIC permite acelerar a execução do treino e classificação visto reduzir-se o número de características de treino (o número de pixels é muito superior ao número de segmentos SLIC). Permite também preservar alguma informação local das regiões das imagens (fundamental para a segmentação), visto um pixel por si não ter informação acerca da região que o rodeia (ao contrário de um segmento SLIC).

Para cada segmento SLIC o programa define um histograma de frequência relativa, representativo dos pixels desse segmento SLIC (em percentagem de ocorrência de cor de pixel). Cada histograma de frequência relativa corresponde às características (vector de valores que caracterizam um objecto) usadas para treinar os classificadores.

Para cada imagem, o programa gera diferentes conjuntos de segmentos SLIC, cada um criado com um conjunto de parâmetros diferentes. A gama de parâmetros SLIC testados é: $n_segments = [2000, 1000, 500]$, $compactness = [20, 10]$, $sigma = [3, 1]$, $threshold = [0.5]$.

Assim, os conjuntos de treino e teste são convertidos em 12 conjuntos diferentes de dados de treino e teste correspondentes à extração de características com as 12 combinações possíveis de parâmetros SLIC. Cada conjunto de dados (treino, teste ou imagem singular para segmentar), é portanto, representado por uma tabela em que as linhas correspondem aos segmentos SLIC de todas as imagens do conjunto, e as colunas correspondem aos histogramas de frequência da cor dos pixels de cada segmento SLIC (linha). No caso dos conjuntos de treino e teste, existe ainda uma última coluna,

que representa a classe de cada segmento SLIC (líquen ou fundo), produto da classificação manual. Na Figura 3.7 observa-se a extração de características para uma imagem.

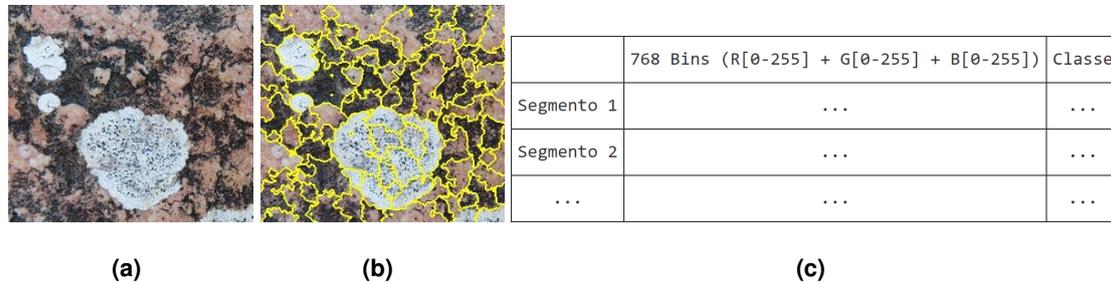


Figura 3.7: Extração de características das imagens.

3.4.2 Determinação dos hiperparâmetros dos classificadores

Quer o classificador *SVC* quer o *RandomForestClassifier* têm várias combinações de parâmetros que podem afectar o desempenho da classificação:

- *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']

A melhor combinação de parâmetros para cada conjunto de dados é obtida com base numa avaliação inicial normalmente chamada estimação de hiperparâmetros.

Para isso, usa-se validação cruzada com 5 *folds* (como descrito na Seção 2.3.3), aplicados aos 12 conjuntos derivados dos parâmetros SLIC.

São feitos os treinos com as múltiplas combinações de hiperparâmetros dos classificadores, e para cada combinação SLIC, obtêm-se o seu desempenho.

A combinação de parâmetros (para o *SVC* e *RandomForestClassifier*) com melhor desempenho (média dos 12 MCC) é então usada no treino e classificações posteriores.

3.4.3 Treino, classificação e avaliação de desempenho

Depois de determinados os hiperparâmetros a usar por cada um dos classificadores, são treinados 24 classificadores usando o conjunto de treino. Estas 24 versões correspondem às instanciações dos classificadores *SVC* e *RandomForestClassifier* com os hiperparâmetros definidos anteriormente, combinado com 12 alternativas de configuração do SLIC (como se mostra na Tabela 3.1).

É importante referir que não é necessário efectuar normalização dos dados de treino, em pré-processamento, pois as características fornecidas aos classificadores são histogramas de frequências relativas que por si só já estão limitadas ao intervalo entre 0 e 1.

Classificador	Hiperparametros	SLIC threshold	SLIC segments	SLIC compact	SLIC sigma
RandomForestClassifier	Melhor combinação de: n_estimators: [150, 100, 50] criterion: [gini, entropy]	0,5	2000	20	3
		0,5	2000	20	1
		0,5	2000	10	3
		0,5	2000	10	1
		0,5	1000	20	3
		0,5	1000	20	1
		0,5	1000	10	3
		0,5	1000	10	1
		0,5	500	20	3
		0,5	500	20	1
		0,5	500	10	3
		0,5	500	10	1
SVC	Melhor combinação de: C: [1,10,100] kernel: [rbf, linear, poly] degree: [2,3,4,5] gamma: [scale, auto] max_iter: [500, 1000]	0,5	2000	20	3
		0,5	2000	20	1
		0,5	2000	10	3
		0,5	2000	10	1
		0,5	1000	20	3
		0,5	1000	20	1
		0,5	1000	10	3
		0,5	1000	10	1
		0,5	500	20	3
		0,5	500	20	1
		0,5	500	10	3
		0,5	500	10	1

Tabela 3.1: Combinação dos parâmetros seleccionados classificadores e SLIC.

Para avaliar o desempenho, o programa segmenta os 12 conjuntos de teste com os dois classificadores, produzindo assim 24 segmentações para cada imagem do conjunto de teste.

A métrica escolhida para efectuar esta avaliação foi o **MCC** (explicado em mais detalhe na Seção 2.3.5). Cada modelo segmenta as imagens do seu conjunto de teste e atribui a cada uma, um valor **MCC**, usando como referencia a classificação manual. O desempenho de cada classificador/parâmetros é obtido calculando a média dos **MCC** para o conjunto de teste. Este indicador de desempenho para cada uma das 24 combinações de classificadores e parâmetros permite ao utilizador o seguinte: decidir adicionar mais imagens ao conjunto de treino e repetir o processo de treino e classificação ou escolher uma das combinações e usar os resultados da classificação correspondente.

Depois de treinados e avaliados, é possível escolher o classificador que poderá produzir melhores resultados com base no **MCC** calculado anteriormente. Este classificador processa as imagens restantes do conjunto de dados e produz as classificações correspondentes.

3.4.4 Processo de escolha de classificador e parâmetros

Esta secção descreve avaliações preliminares feitas ao programa de classificação automática.

No decorrer deste trabalho foram utilizados seis conjuntos de imagens listados na Tabela 3.2 com o intuito de testar as funcionalidades do programa. Os conjuntos 1, 2 e 3 resultaram da selecção de algumas imagens de conjuntos maiores, listados na Tabela 4.2 e descrito na Seção 4.1. As imagens pertencentes aos conjuntos não foram adquiridas com o sistema de miras. Os conjuntos de dados 1

Nome	Local	Observações	nº imagens	Tamanho (pixels)
dataset 1	Nazaré	Imagens retiradas do conjunto de dados Nazaré 3	10	3456 x 5184
dataset 1 small	Nazaré	Igual ao dataset 1 mas as imagens foram reduzidas para 6% das originais	10	208 x 312
dataset 2	Cascais	Imagens retiradas do conjunto de dados Cascais	14	2000 x 3008
dataset 2 small	Cascais	Igual ao dataset 2 mas as imagens foram reduzidas para 10% das originais	14	200 x 301
dataset 3	Antárctida	Imagens retiradas do conjunto de dados Terraço	15	3888 x 5184
dataset 3 small	Antárctida	Igual ao dataset 3 mas as imagens foram reduzidas para 10% das originais	15	389 x 519

Tabela 3.2: Mini-conjuntos de dados utilizados no desenvolvimento do programa.

small, *2 small* e *3 small* são idênticos aos seus homólogos mas com resolução das imagens reduzida, de forma a reduzir os tempos de execução dos testes.

Isto foi necessário para avaliar diferentes tipos de características já que, por exemplo, ao usar directamente pixels como características o número das mesmas torna-se muito elevado para imagens com muita resolução e, conseqüentemente, aumenta de forma significativa os tempos de execução do programa.

Estes conjuntos serviram para testar quais os classificadores e características que deram melhores resultados (desempenho e qualidade das segmentações), assim como para provar que as características SLIC com histogramas de frequência relativa apresentam bons resultados e com tempos de execução aceitáveis para imagens de resolução alta. Também serviram para encontrar e testar os intervalos de valores dos parâmetros dos classificadores e características SLIC a utilizar na versão final do programa. É de relembrar que, tanto a função *Gridsearch* como as características SLIC, são executados com listas de parâmetros diferentes. Também se compararam os resultados com imagens representadas nos modelos de cor RGB e HSV, para ver se alguns destes apresentaria melhores segmentações face ao outro.

Foram feitos dois tipos de avaliações/testes preliminares: (i) selecção do modelo e características, e (ii) avaliação de parâmetros SLIC. Para o primeiro tipo, foram utilizados os conjuntos de imagens com menor resolução para reduzir os elevados tempos de execução. Este teste consiste em treinar vários classificadores, cada um com uma lista de hiperparâmetros possíveis, e usando diferentes tipos de características. Os classificadores testados foram: *GaussianNB*, *DecisionTreeClassifier*, *RandomForestClassifier* e *SVC*.

As características testadas foram: características pixel (em que cada pixel é transformado nas suas componentes RGB), características SLIC (como explicado na Seção 2.3.4) e *nn.layer* (em que as características são a resposta às imagens de entrada da primeira camada convolucional da rede VGG16 com pesos pré-treinados no conjunto de imagens da *imagenet* (Figura 3.8)).

Procurou-se encontrar os pares classificador-características que apresentaram melhores resultados e desempenho (medido no conjunto de teste em precisão e tempo de execução). O programa foi testado três vezes para cada conjunto de imagens de resolução reduzida e para cada teste foram escolhidas diferentes imagens de treino e teste. Ou seja, no total, foram efectuados nove testes, em que para cada conjunto de imagens de resolução reduzida foram efectuados três testes com conjuntos de treino

e teste diferentes.

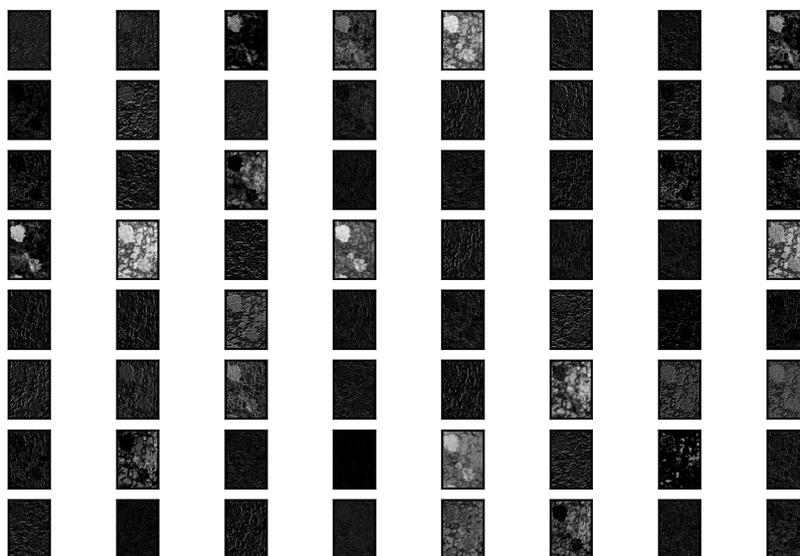


Figura 3.8: Resposta da primeira camada convolucional da rede VGG16 (pesos pré-treinados na *imagenet*) a uma imagem de líquen.

Em cada teste são avaliadas todas as combinações classificador-características. No entanto, como as características SLIC têm vários parâmetros, a avaliação requer efectuar um teste para cada um deles, o que torna os resultados não balanceados no que toca ao número de testes efectuados para cada par classificador-características, sendo que haverá muito mais testes com características SLIC comparativamente às restantes características.

De forma a balancear os resultados, apenas se consideraram os melhores resultados de cada conjunto classificador-características. Desta forma, garante-se que não existe tendência para o tipo de testes mais numeroso.

Em seguida, escolheram-se os cinco que obtiveram melhor precisão no conjunto de imagens de teste para cada exemplo, listados na Tabela A.1 em anexo.

É importante notar que a precisão não é uma métrica ideal na classificação binária quando as classes não são balanceadas. Por exemplo, dada uma imagem com apenas 5% de pixels pertencentes à classe líquen e 95% pertencentes à classe fundo, se o programa classificar a imagem toda como sendo fundo, então o erro foi apenas de 5% e a precisão foi de 95%, o que à primeira vista parece ser um bom resultado, no entanto, o programa nem sequer detectou o líquen presente na imagem. Ou seja, a precisão não é uma boa métrica quando as classes não são balanceadas. De facto, observa-se

que o conjunto de imagens de resolução reduzida 3, em particular, apresenta uma grande disparidade nas duas classes.

Da análise dos resultados deste primeiro teste, conclui-se que os modelos de classificadores que apresentaram consistentemente melhor precisão face aos restantes, foram o *SVC* e o *RandomForestClassifier*. Também se concluiu que as melhores características foram as características *SLIC*, tanto, na precisão como no tempo de execução e que, apesar das características *pixel* e *nn_layer* apresentarem boa precisão nos conjuntos de teste, os requisitos computacionais são muito elevados. Para estes casos, os tempos de execução foram demasiado elevados, o que não conduz a uma boa escalabilidade do sistema (Tabela A.1 em anexo). Estes problemas ir-se-ão intensificar nos casos em que se faça uso de um maior número de imagens no treino, assim como na classificação de imagens de alta resolução.

Para o segundo teste decidiu-se apenas utilizar o modelo de cor *RGB* já que, como seria de esperar, não se verificou nenhuma diferença significativa nos resultados entre os modelos de cor *RGB* e *HSV*. O segundo tipo de testes (ver Tabela A.2 em anexo) serviu para avaliar e validar o uso das características *SLIC* com histogramas de frequência relativa, assim como para descobrir quais as gamas de valores dos parâmetros *SLIC* mais indicados, para a segmentação dos líquenes. Neste caso, também foram efectuados três testes para cada um dos conjuntos de imagens 1, 2, e 3, cada vez com imagens de treino e teste diferentes. No entanto, os classificadores utilizados foram apenas o *RandomForestClassifier* e *SVC*. Verificou-se que o tipo de classificador e parâmetros *SLIC* que produzem os melhores resultados estão fortemente dependentes do conjunto em questão. Ou seja, para diferentes imagens e conjuntos, não se consegue saber à partida quais os pares classificador-parâmetros *SLIC* ideais. Para tal, tem que se testar, para cada imagem, os dois classificadores e um conjunto de parâmetros *SLIC* de forma a maximizar a qualidade das segmentações. É de notar que, como se usaram poucas imagem de treino e teste, é possível que estes resultados apresentem uma variância elevada pois dependem muito dos conjuntos de imagens específicos utilizados no treino e teste.

Na Tabela A.2 apresentam-se os cinco melhores casos de cada teste. Neste caso já não se usou a precisão como métrica mas sim a média das taxas dos verdadeiros positivos e negativos obtidos nos conjuntos teste de cada exemplo.

Na Tabela 3.3 podemos observar com mais detalhe os resultados individuais das imagens do conjunto de teste número 9 (relembro que o conjunto de dados 3 *small* apresenta grandes disparidades no balanço das classes). O parâmetro *L_ratio* é a percentagem de pixels da imagem que pertencem a líquenes. Quanto este valor é muito baixo, então a percentagem de líquenes presente na imagem é muito baixa comparativamente ao resto. No entanto, verifica-se que a precisão se mantém sempre nos 99% mesmo em situações em que a taxa de verdadeiros positivos é mais baixa, ou seja, a precisão não é a métrica ideal nesta situação.

Após estes testes e observações dos vários resultados, decidiu-se usar no sistema os algoritmos e

Teste nº	Imagem	L ratio	Precisão	Taxa de positivos verdadeiros	Taxa de negativos verdadeiros
9	IMG_1947.JPG	0.00817	0.9971	0.8408	0.9984
			0.9979	0.7588	0.9999
			0.9978	0.7558	0.9998
			0.9963	0.5614	0.9999
			0.9983	0.8093	0.9998
	IMG_1983.JPG	0.01527	0.9863	0.9612	0.9867
			0.9969	0.8515	0.9992
			0.9967	0.8258	0.9994
			0.9940	0.6671	0.9991
			0.9921	0.4945	0.9997
	IMG_1928.JPG	0.02996	0.9802	0.9754	0.9803
			0.9940	0.9107	0.9965
			0.9949	0.9029	0.9977
			0.9953	0.8926	0.9985
			0.9934	0.8080	0.9991

Tabela 3.3: Detalhes dos resultados do teste nº9 para cada uma das três imagens do conjunto de teste.

parâmetros enumerados na tabela apresentada na Seção 3.4.3. Ou seja, para cada imagem serão produzidas 24 segmentações correspondentes às combinações dos dois classificadores com parâmetros SLIC.

3.5 Cálculo das áreas dos líquenes

Os resultados das classificações apenas identificam as áreas ocupadas por líquenes, mas é fundamental que outros dados sejam extraídos, tal como o número de talos e área de cada um.

Estes cálculos são efectuados por um programa que recebe as imagens binárias das segmentações (imagens a preto e branco), e para cada líquen, atribui um índice, retorna a área, *filled_area* (número de pixels na região com todos os buracos preenchidos ver líquen de índice 7 da Figura 3.9 e Tabela 3.4), perímetro e coordenadas do centróide.

O programa utiliza a função *regionprops* da biblioteca *measure* do *skimage* para efectuar a análise das regiões das imagens.

Se o utilizador atribuir uma escala à imagem, é possível converter a área de cada líquen de pixels para mm².

A escala pode ser retirada automaticamente a partir da imagem, caso tenha sido capturada com o sistema de miras, ou manualmente se a imagem incluir outro tipo de instrumento de medição (régua, esquadro).

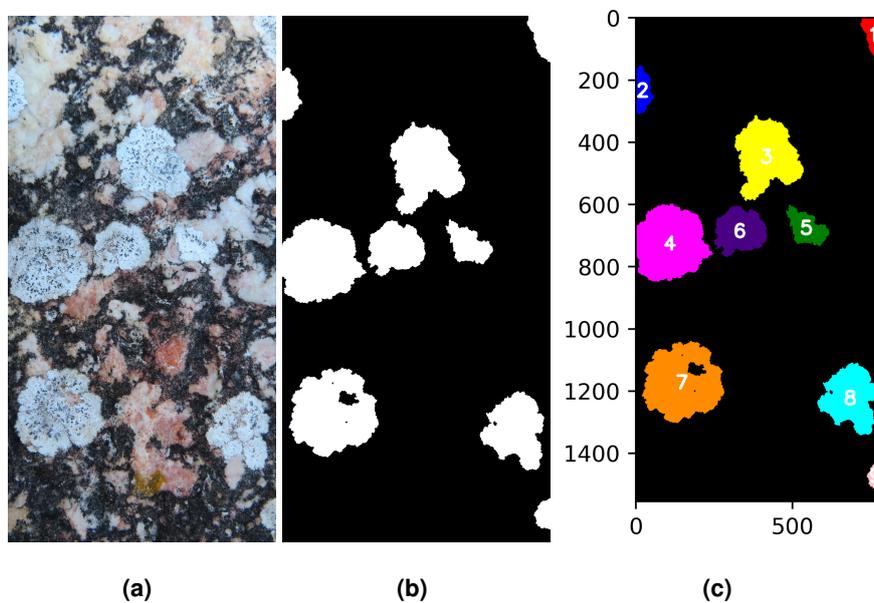


Figura 3.9: Exemplo de medição de líquenes numa imagem sem escala, os eixos representam a altura e largura da imagem em pixels.

Índice do líquen	Centroid(pixels)	Area(pixels)	Filled area(pixels)	Perimeter(pixels)
1	(54.79, 759.2)	6414	6414	380,27
2	(233.6, 21.48)	6251	6251	394,69
3	(446.5, 416.44)	37623	37623	1079,38
4	(725.67, 108.72)	45021	45021	992,93
5	(675.31, 544.56)	10118	10118	542,16
6	(685.64, 331.98)	19475	19475	685,39
7	(1172.27, 147.42)	47611	49335	1348,92
8	(1222.79, 684.33)	27967	28063	930,53
9	(1473.06, 765.89)	3253	3253	258,75

Tabela 3.4: Resultado das medições para a Figura 3.9(a).

4

Avaliação

Conteúdo

4.1	Descrição dos conjuntos de dados e ambiente de teste	37
4.2	Sistema de miras	38
4.3	Processamento de imagens	40
4.4	Resultados das classificações	44

Neste capítulo irá ser feita a avaliação do desempenho das diferentes componentes que comportam este trabalho. Nomeadamente, o sistema de miras, o método de classificação manual com o GrabCut, o uso das características SLIC/histogramas de frequência relativa, a classificação automática e os seus resultados. Em relação à classificação automática, também será elaborado um estudo de forma a analisar as curvas de aprendizagem, a escalabilidade e desempenho dos modelos de classificação. Da mesma forma, os resultados finais obtidos da segmentação automática serão analisados.

4.1 Descrição dos conjuntos de dados e ambiente de teste

Todo o código foi desenvolvido e testado numa máquina com as especificações e ambiente de desenvolvimento indicados na Tabela 4.1.

Hw Servidor	Hw Desenvolvimento	Bibliotecas
Ubuntu 18.04.4	Windows 10	Python 3.7.9
Arquitetura 64 bits	Arquitetura 64 bits	Pandas 1.2.1
RAM: 15GB	RAM: 8 GB	Anaconda 1.9.12
CPU: Intel(R) Xeon(R) CPU E3-1230 v5 @ 3.40GHz 8 Cores	CPU: Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz 4 Cores	Opencv 4.5.1
		Scikit-learn 0.24.1
		Scikit-image 0.17.2
		Spyder 5.0.0

Tabela 4.1: Especificações do hardware e ambiente utilizados no desenvolvimento do trabalho e testes.

Foi também utilizado um servidor para efectuar alguns testes devido aos elevados tempos de execução. Apenas os códigos de classificação automática e estudo das características é foram executados no servidor.

As bibliotecas de aprendizagem automática utilizadas não fizeram uso das capacidades gráficas, GPU's, por isso essa informação foi omitida.

Os conjuntos de dados presentes na Tabela 4.2 foram utilizados para avaliar o programa de classificação automática.

Nome	Local	Líquenes	Características	Rocha	Nº imagens	Tamanho (pixels)
Terraço	Antártida	<i>Buellia latemarginata</i> , <i>Rhizocarpon geographicum</i>	Branços com apotécios pretos, Verdes	Argillito/siltito	63	3888 x 5184
Nazaré 1	Nazaré	<i>Opegrapha durieui</i>	Branços	Calcário	27	1944 x 2592
Nazaré 2	Nazaré	<i>Opegrapha durieui</i>	Branços	Calcário	40	3456 x 4608
Nazaré 3	Nazaré	<i>Opegrapha durieui</i>	Branços	Calcário	52	3456 x 5184
Muro Castelejo 2	Fundão	<i>Rhizocarpon sp.</i> , Espécie desconhecida	Verdes, Branços	Xisto	38	3456 x 5184
Muro Escola Castelejo	Fundão	<i>Rhizocarpon sp.</i> , <i>Rhizocarpon</i>	Verdes, Pretos	Granito	17	3456 x 5184
Cascais	Cabo Raso	<i>Opegrapha durieui</i> , <i>Caloplaca sp.</i>	Branços, Laranjas	Calcário	63	2000 x 3008

Tabela 4.2: Conjunto de dados usados.

4.2 Sistema de miras

De modo a avaliar o sistema de miras, foram efectuadas experiências com 18 fotografias. As experiências foram realizadas no mesmo local onde se fez a aquisição do conjunto de dados Cascais (Tabela 4.2). Foi utilizada uma câmara de telemóvel com 13 MP de resolução.

O procedimento é semelhante ao que seria feito para amostragem de líquenes, no entanto neste caso utiliza-se um alvo.

Cada experiência consiste em:

1. Colocar o alvo sobre uma superfície rochosas
2. Colocar as miras na mesma superfície de forma a cercarem o alvo
3. Tirar uma fotografia ao alvo com as miras dispostas à volta
4. Endireitar a imagem
5. Medir os eixos maior e menor do alvo
6. Comparar os resultados com os valores de referência

É importante referir que as fotografias foram tiradas sem seguir uma determinada configuração (zoom) e posicionamento da máquina em relação ao objecto.

Tanto as miras como o alvo têm dimensões bem definidas. O conjunto de miras é constituído por 4 círculos azuis colados sobre uma base rectangular e os centros destes círculos formam uma região rectangular com 272mm de comprimento e 185mm de altura (medido com uma régua). O alvo circular tem 60mm de diâmetro medidos experimentalmente com o auxílio de uma craveira. As miras e alvo são fixados às superfícies rochosas através de pasta adesiva.

As medições dos eixos menor e maior do alvo na imagem endireitada (Figura 4.1(b)) foram efectuadas automaticamente através dos parâmetros *major_axis_length* e *minor_axis_length* da função *regionprops* pertencente à biblioteca *skimage* de *python* (Figura 4.1(c)).

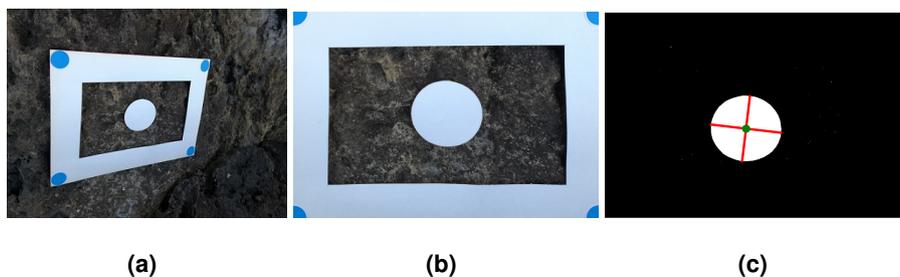


Figura 4.1: Exemplo de medição dos eixos maior/menor do alvo.

É importante referir que apenas não foi possível fazer a detecção das 4 miras numa das experiências. Devido ao céu presente na imagem (Figura 4.2), que possui uma tonalidade semelhante à das miras, não foi possível fazer a detecção, já que o sistema faz a detecção por segmentação de cor (o resultado dessa experiência foi omitido da tabela Tabela 4.3).

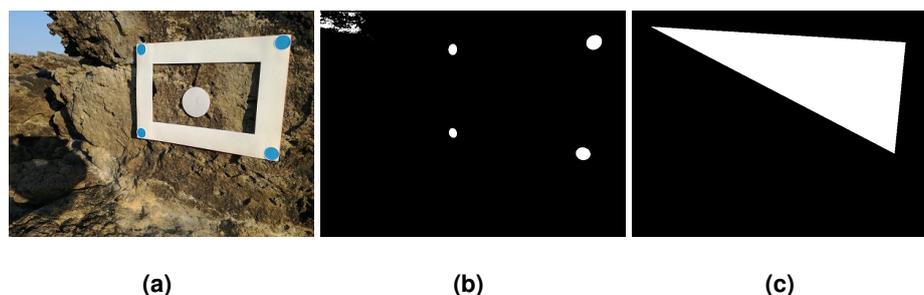


Figura 4.2: Caso de falha da detecção das miras.

Na Tabela 4.3 observam-se os resultados das experiências efectuadas.

Experiência	Eixo A (mm)	Erro A	Eixo B (mm)	Erro B	Área (mm ²)	Erro área
1	58,19	3,02%	57,74	3,77%	2638,58	6,68%
2	57,79	3,68%	53,73	10,45%	2438,32	13,76%
3	61,48	2,47%	55,56	7,40%	2681,77	5,15%
4	61,88	3,13%	58,54	2,43%	2844,95	0,62%
5	59,3	1,17%	59,17	1,38%	2755,85	2,53%
6	62,92	4,87%	59,03	1,62%	2916,88	3,16%
7	62,93	4,88%	55,02	8,30%	2718,2	3,86%
8	62,89	4,82%	54,59	9,02%	2695,56	4,66%
9	58,76	2,07%	57,87	3,55%	2670,75	5,54%
10	57,97	3,38%	53,96	10,07%	2456,52	13,12%
11	59,64	0,60%	55,48	7,53%	2598,21	8,11%
12	62,46	4,10%	57,84	3,60%	2837,59	0,36%
13	61,89	3,15%	56,67	5,55%	2754,33	2,59%
14	64,59	7,65%	54,12	9,80%	2744,74	2,92%
15	60,86	1,43%	50,36	16,07%	2406,41	14,89%
16	55,9	6,83%	49,74	17,10%	2183,05	22,79%
17	56,81	5,32%	55,34	7,77%	2469	12,68%
Referência:	60		60		2827,43	

Tabela 4.3: Resultados das experiências efectuadas com as miras.

Nota-se que apesar do erro da área ter atingido, em certas experiências, valores da ordem dos 20%, também se obteve um considerável número valores baixos, da ordem dos 5%, sendo que a média do erro obtido foi de 7.26%. Este erro provém essencialmente do facto do alvo e as miras não fazerem parte do mesmo plano, já que as rochas são irregulares, assim como possíveis erros de medição do raio do alvo e distâncias entre as miras. Outra possível causa é a distorção da lente da câmara digital.

4.3 Processamento de imagens

Nesta secção vai-se avaliar o desempenho do GrabCut no que se refere ao tempo que o utilizador leva a classificar as fotografias e qual o erro que esta aproximação tem quando comparado com o processo habitual. Para se avaliar o processamento das imagens com o GrabCut ir-se-á medir o tempo que leva a classificação com o GrabCut e o erro dessa classificação face às técnicas descritas na Secção 2.1.1.

Ir-se-á também avaliar o erro existente na conversão das imagens para segmentos SLIC (fronteiras dos segmentos SLIC nem sempre coincidem com os limites das superfícies de líquenes), assim como o tempo de execução da transformação das imagens de entrada em características para uso no programa de classificação automática.

4.3.1 Classificação manual

O uso do GrabCut permite reduzir o tempo que o utilizador despende para classificar as imagens. Esta nova ferramenta é usada para a criação dos conjuntos de imagens de treino/teste, mas pode também ser usada de forma isolada, substituindo os processamentos até agora usados.

A avaliação do desempenho do GrabCut foi efectuada comparando as classificações produzidas com identificação manual e semi-automática de líquenes, com recurso às aplicações ArcGIS™ e Photoshop® como referência, no que se refere ao tempo e diferenças de classificação.

Para a realização desta avaliação foi pedido a um investigador que classificasse 18 fotografias: 9 foram classificadas com o Photoshop® e com o GrabCut e outras 9 foram classificadas com o ArcGIS™ e com o GrabCut. Os resultados e características, ganhos (de tempo) e precisões estão apresentadas na Tabela 4.4.

Conjunto de dados	Imagem	Resolução	Método	Tempo (min)	Tempo GrabCut (min)	Ganho	Precisão do GrabCut	MCC do GrabCut
Cascais	IMG.8522	5184 x 3456	Vectorização com ArcGIS	19	2	89%	99,85%	0,9684
Forte Sta Susana 2	IMG.6783	5184 x 3456	Vectorização com ArcGIS	14	6	57%	98,33%	0,9151
Muro Castelejo 2	IMG.7653	5184 x 3456	Vectorização com ArcGIS	8	5	38%	98,00%	0,9594
Muro Escola Castelejo	IMG.7591	5184 x 3456	Vectorização com ArcGIS	2	2	0%	99,58%	0,9702
Muros Aldeia Trebilhadouro	IMG.7587	5184 x 3456	Vectorização com ArcGIS	5	1	80%	98,48%	0,9446
Nazare 3	IMG.6678	5184 x 3456	Vectorização com ArcGIS	20	3	85%	99,11%	0,9624
Portela da Anta	IMG.7417	5184 x 3456	Vectorização com ArcGIS	6	3	50%	99,64%	0,9274
Queda de Bloco	IMG.6591	5184 x 3456	Vectorização com ArcGIS	37	4	89%	98,85%	0,9255
Terraço Antartida	IMG.1912	5184 x 3888	Vectorização com ArcGIS	10	3	70%	99,77%	0,9002
Cascais	IMG.8573	5184 x 3456	Classificação Photoshop	28	9	68%	98,63%	0,9609
Forte Sta Susana 2	IMG.6821	5184 x 3456	Classificação Photoshop	31	9	71%	96,60%	0,9309
Muro Castelejo 2	IMG.7628	5184 x 3456	Classificação Photoshop	24	3	88%	99,07%	0,8970
Muro Escola Castelejo	IMG.7603	5184 x 3456	Classificação Photoshop	26	3	88%	98,04%	0,8142
Muros Aldeia Trebilhadouro	IMG.7578	5184 x 3456	Classificação Photoshop	15	1	93%	99,49%	0,9322
Nazare 3	IMG.6683	5184 x 3456	Classificação Photoshop	124	9	93%	94,95%	0,8350
Portela da Anta	IMG.7444	5184 x 3456	Classificação Photoshop	55	3	95%	95,77%	0,8482
Queda de Bloco	IMG.6585	5184 x 3456	Classificação Photoshop	33	2	94%	99,36%	0,8995
Terraço Antartida	IMG.1928	5184 x 3888	Classificação Photoshop	27	5	81%	99,49%	0,9225
Média:						74%	98,50%	0,9174

Tabela 4.4: Tempos e erros medidos na avaliação das segmentações manuais.

Pode observar-se que o GrabCut é sempre mais rápido que o método com Photoshop® e ArcGIS™,

e ainda verificar-se que numa série de experiências se obteve um ganho da ordem dos 90%. Ou seja, nesses casos, o utilizador necessita apenas despender de 10% do tempo que teria de despender com os métodos manuais.

A Tabela 4.4 apresenta os erros associados obtidos para todas as experiências. Esta tabela compara os resultados do GrabCut face ao resultados obtidos com o Photoshop® e ArcGIS™, escolhidos como referências. Esta comparação será efectuada em termos de precisão e MCC.

Observa-se que a precisão está sempre acima dos 95% (na grande maioria dos casos, acima de 98%) e que o MCC em média apresenta um valor de 0.9174. Note-se que o MCC é consistentemente ligeiramente inferior a precisão, isto acontece porque o MCC é uma métrica mais precisa neste contexto de classificação binária, já que atribui maior peso á correcta classificação de ambas as classes e não quantifica apenas um valor absoluto de diferenças na classificação.

De modo a analisar com mais detalhe os resultados do MCC, traçou-se o gráfico representado na Figura 4.3, que ilustra a relação do MCC com o tempo de classificação manual do ArcGIS™/Photoshop®.

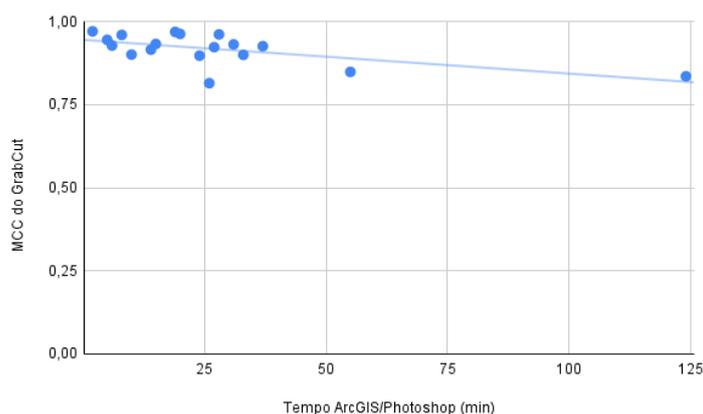


Figura 4.3: Evolução do erro do GrabCut com o tempo de classificação manual do ArcGIS™ e Photoshop®.

Verifica-se que existe uma tendência, dado que quanto mais complexa/difícil de segmentar é uma imagem, maior é o erro da classificação efectuada com o GrabCut (Figura 4.3).

4.3.2 Avaliação das características (SLIC e histogramas)

Como o SLIC faz *clustering* de áreas das imagens, pode introduzir erro, por as fronteiras dos segmentos SLIC nem sempre coincidirem completamente com as superfícies dos líquenes. Ou seja, pode acontecer que um dado segmento SLIC contenha pixels de fundo e de líquen.

Nesta secção vai-se avaliar a pertinência do uso do SLIC e histogramas correspondentes no treino dos algoritmos, de modo a quantificar o erro associado.

Uma avaliação preliminar já foi apresentada na Seção 3.4.4, onde se mostram as vantagens do uso das características SLIC face a outras, e se verificam melhorias nos tempos de execução e de classificação. No entanto essa análise foi efectuada comparativamente a outros tipos de características. Contrariamente, nesta secção, pretende-se avaliar o erro absoluto e os tempos de execução.

Para tal, foram utilizadas as mesmas 18 imagens da secção anterior, e as classificações binárias (efectuadas manualmente) correspondentes. No entanto, foram reduzidas para diferentes resoluções (40%, 35%, 30%, 25%, 20%, 15% e 10%) das originais, de modo a agilizar o processo de avaliação e determinar a escalabilidade temporal destas características.

Os parâmetros SLIC testados foram os mesmos utilizados na classificação automática: $n_segments = [2000, 1000, 500]$, $compactness = [20, 10]$, $\sigma = [3, 1]$, $threshold = [0.5]$.

Foram testadas as 18 imagens para as 12 combinações de parâmetros SLIC, correspondendo a 216 testes, sendo que estes foram repetidos para as mesmas imagens mas com diferentes resoluções.

Cada teste consiste em aplicar o *clustering* SLIC a cada imagem, atribuir uma classe a cada segmento SLIC (através da comparação com a classificação binária correspondente, feita com o Photoshop®/ArcGIS™) e comparar o resultado com a referência. Ou seja, utilizaram-se as classificações binárias tanto como referência, como para atribuir a classe a cada segmento SLIC.

É importante lembrar que é na atribuição de classe a cada segmento que entra o parâmetro *threshold* do SLIC. Por exemplo para um valor de 0.5 (50%), se um dado segmento contem mais de metade de pixels de líquen (sabe-se comparando com a classificação feita manualmente) então atribui-se a classe de líquen a esse segmento, caso contrario, atribui-se a classe fundo.

No fim, compararam-se as diferenças e mediram-se os tempos. Os tempos de geração das características correspondem à soma do tempo de criação dos segmentos SLIC, o tempo de converter cada segmento SLIC no histograma de frequência relativa correspondente (conforme as cores dos pixels dessa região) e, no caso de serem dados de treino, o tempo de atribuição de classe a cada segmento.

Na Tabela 4.5 observam-se os resultados médios obtidos para cada conjunto de imagens com resoluções diferentes.

Redução	Resolução	Precisão	MCC	SLIC (sec)	Histogramas (sec)	Atribuir classe (sec)
40%	2073 x 1382	97,93%	0,8633	3,84	27,51	3,43
35%	1814 x 1209	97,99%	0,8663	2,93	21,42	2,50
30%	1555 x 1036	98,00%	0,8679	2,20	15,80	2,70
25%	1296 x 864	98,03%	0,8704	1,41	10,52	1,09
20%	1036, x 691	98,03%	0,8706	0,90	6,41	0,63
15%	777 x 518	98,03%	0,8699	0,51	3,53	0,32
10%	518 x 345	97,94%	0,8623	0,23	1,72	0,15

Tabela 4.5: Resumo dos resultados médios obtidos nos testes das características SLIC e histogramas para imagens de resoluções diferentes.

Observa-se que os valores médios da precisão e do MCC são relativamente constantes com variação

da resolução das imagens, a precisão sempre na ordem dos 98% e MCC nos 0.87. Mais uma vez, verifica-se que o MCC apresenta consistentemente valores inferiores á precisão.

Nos casos em que o tamanho dos segmentos SLIC (dado pelo parâmetro *n_segments*) é superior às regiões dos líquenes, o programa pode não conseguir converter as regiões com líquenes presentes nas imagens para as suas características, originando segmentos de características atribuídos apenas com a classe de fundo. O valor da precisão nesses casos pode apresentar valores altos (correspondentes à percentagem do fundo presente na imagem), no entanto a métrica MCC detecta esses casos originando $MCC=0$.

O gráfico da Figura 4.4 ilustra a relação entre os tempos e a resolução.

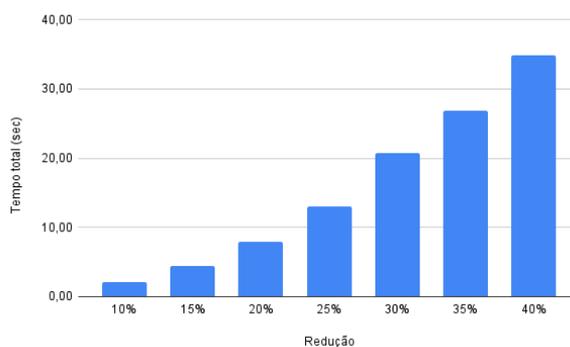


Figura 4.4: Relação entre os tempos de execução totais (médios) e a resolução das imagens.

Verifica-se que os resultados dos tempos aumentam com a resolução das imagens. Ao analisar os resultados do gráfico, verifica-se que os tempos são mais elevados quanto maiores for a resolução das imagens (mais pixels a serem processados). Para além disso, o tempo de cálculo dos histogramas é maior quanto maior o número de segmentos SLIC, dado ser necessário calcular um maior numero de histogramas (cada segmento SLIC é convertido num histograma correspondente). É de notar, que o cálculo dos histogramas é o que tem maior peso no tempo de execução.

No entanto, o mais importante a ter em conta nestes resultados, é o erro da conversão para segmentos SLIC. O principal objectivo é minimizar este erro, já que se que usam estas características baseadas em segmentos SLIC para treinar os classificadores. Este erro deve-se ao facto das fronteiras das regiões SLIC nem sempre coincidirem com as delimitações das superfícies dos líquenes. Na Figura 4.5 observa-se a distribuição dos resultados (dados pelo MCC) para o caso dos testes efectuados ao conjunto de imagens reduzidas para 30% e verifica-se que, em geral, o erro é baixo (valores altos MCC) sendo que essa distribuição particular apresenta média de 0.8679 e um desvio padrão de 0,1327.

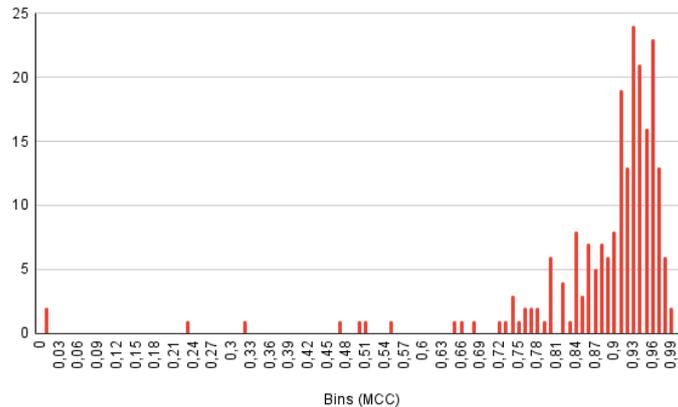


Figura 4.5: Distribuição dos resultados do MCC para imagens com resolução 1555 x 1036.

4.4 Resultados das classificações

Esta secção apresenta a avaliação do comportamento e respostas dos algoritmos de classificação automática para conjuntos de imagens variados. Pretende-se com isto, testar a robustez do sistema desenvolvido e verificar a capacidade de generalização do problema, com a análise da qualidade das segmentações produzidas.

Nesta secção, foram utilizados os 7 conjuntos de imagens descritos na Tabela 4.2, para o treino/teste e avaliação do programa de classificação automática. As imagens destes conjuntos foram reduzidas para 15% das originais de modo a agilizar o processo de avaliação.

4.4.1 Curvas de aprendizagem e escalabilidade do sistema

De modo a medir a capacidade de aprendizagem e escalabilidade dos algoritmos, foram efectuados vários ciclos de treino, para cada um dos 7 conjuntos de imagens, cada um com um número incremental de imagens de treino. No fim de cada ciclo de treino, são segmentadas as imagens do conjunto de teste e comparadas às de referência (classificações manuais com o GrabCut) de modo a verificar a qualidade das segmentações (com a métrica MCC) desse ciclo. A escalabilidade foi estudada analisando o tempo de execução de cada ciclo.

Como descrito nas secções anteriores em cada ciclo de treino, os dois classificadores (*Random-ForestsClassifier* e *SVC*) são treinados para cada combinação dos 12 parâmetros SLIC, originando na prática a 24 ciclos de treino. Devido às características desta avaliação, que implica a execução de vários ciclos de treino, e que em cada um o número de imagens de treino é incrementado (o que aumenta ainda mais os tempos de execução) foi decidido abandonar a validação cruzada. Esta decisão foi necessária de modo a manter os tempos de execução em valores razoáveis.

Dado isso foram utilizados os parâmetros padrão dos classificadores. Para o *RandomForestsClassifier* foram: 'n_estimators': [100], 'criterion': ['gini']. Para o SVC foram: 'C': [1], 'kernel': ['rbf'], 'gamma': ['scale'], 'max_iter': [-1]

Na Figura 4.6 estão ilustrados os resultados obtidos para os conjuntos de imagens da Antártida e Cascais (os resultados para os outros conjuntos de dados estão no Figura A.2). Cada gráfico comporta informação das 12 combinações possíveis de parâmetros SLIC para um classificador individual.

Observa-se que as curvas de aprendizagem têm tendência a aumentar com o aumento do número de imagens utilizado em cada ciclo de treino. Ou seja, quantas mais imagens forem utilizadas no treino melhor será a qualidade das classificações.

Comparando as várias curvas de aprendizagem conclui-se que, dependendo do conjunto de imagens, os parâmetros SLIC podem ou não ter impacto na qualidade das segmentações. Isto deve-se à variância intrínseca presente em cada conjunto de imagens. É também importante notar que os classificadores podem apresentar diferenças na qualidade das segmentações, sendo que em geral, o SVC parece apresentar resultados superiores face ao *RandomForestsClassifier*.

Analogamente, verifica-se que o tempo de treino de ambos os classificadores aumenta com o número de imagens de treino, ou seja, quanto mais imagens se utilizam mais tempo se leva a efectuar o treino.

Nos gráficos dos tempos de treino, verificam-se semelhanças nas curvas com o mesmo parâmetro SLIC *n_segments*. Isto quer dizer que o tempo de treino está fortemente ligado a este parâmetro e que quantos mais segmentos SLIC foram gerados, maior será o número de características dos dados de treino, o que obriga os classificadores a processar mais informação, aumentando o tempo de treino e piorando a escalabilidade dos algoritmos.

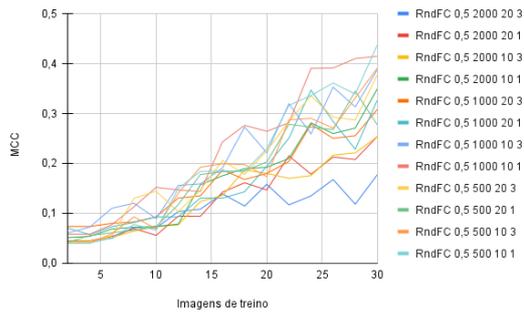
Verifica-se também que os tempos de treino dos classificadores *RandomForestsClassifier* evoluem linearmente no tempo, ao contrário dos SVC que evoluem exponencialmente. Também se pode observar que, no caso do conjunto de imagens de Cascais, o SVC apresenta tempos de treino superiores ao *RandomForestsClassifier*. Logo o *RandomForestsClassifier* apresenta melhor escalabilidade face ao SVC.

Observa-se que os valores de MCC estabilizam a partir de 20-25 imagens de treino. É uma conclusão muito importante para definir metodologias de melhor custo-benefício.

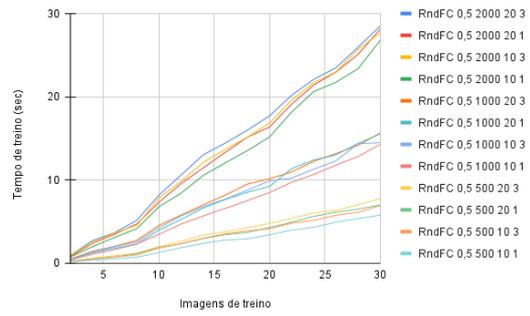
4.4.2 Análise das segmentações

Nesta secção, analisaram-se algumas segmentações de modo a ilustrar limitações do programa desenvolvido, assim como o potencial existente neste tipo de segmentações.

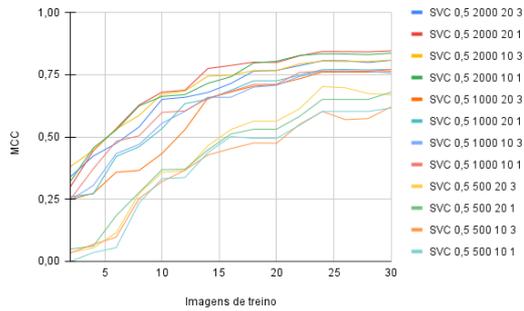
Foram escolhidas para análise algumas segmentações de modo a ilustrar certas características das segmentações. Pretende-se portanto avaliar a qualidade das segmentações produzidas especifica-



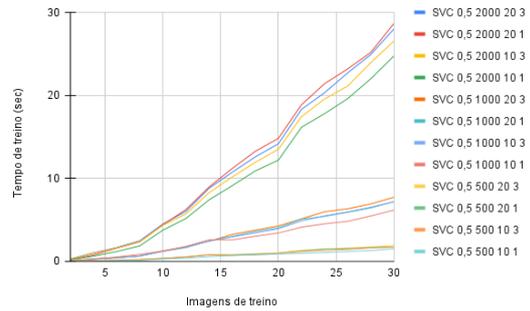
(a) Antártida



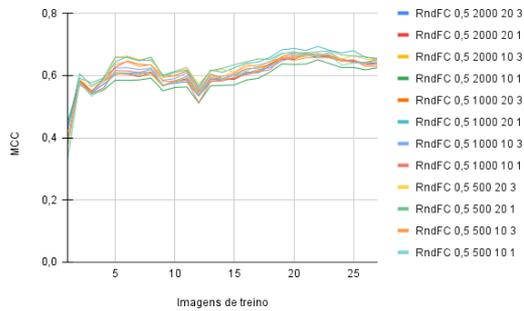
(b) Antártida



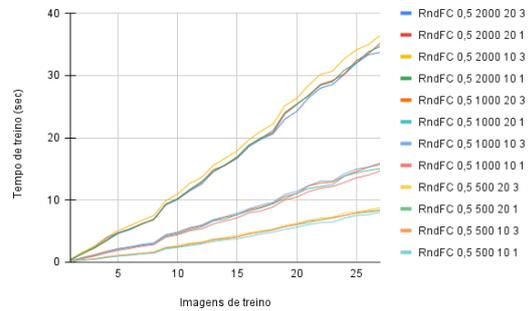
(c) Antártida



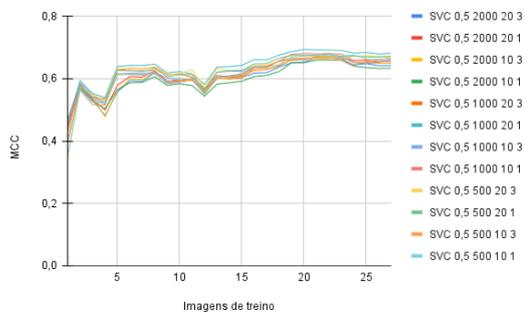
(d) Antártida



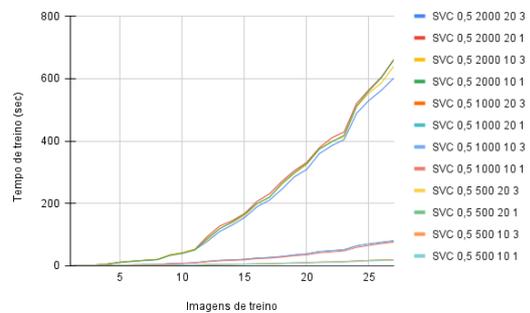
(e) Cascais



(f) Cascais



(g) Cascais



(h) Cascais

Figura 4.6: Gráficos das curvas de aprendizagem e escalabilidade para os conjuntos de imagens da Antártida e Cascais.

mente no que se refere aos verdadeiros/falsos positivos/negativos.

No seguimento da secção anterior, são apresentadas algumas segmentações do conjunto da Antártida, Cascais e também do conjunto Muro Escola Castelejo (Figura 4.7). As imagens estão dispostas de forma a que, em cada linha, se observa primeiro a imagem original, em seguida a classificação manual e por último uma classificação automática.

As classificações automáticas representadas na Figura 4.7(c) e Figura 4.7(f) (Antártida) foram produzida pelo classificador SVC, treinado com 30 imagens e com os seguintes parâmetros SLIC: *threshold* 0.5, *n_segments* 500, *compactness* 20 e *sigma* 1. A classificação automática representada na Figura 4.7(i) (Cascais) foi produzida pelo classificador *RandomForestsClassifier*, treinado com 27 imagens e com os seguintes parâmetros SLIC: *threshold* 0.5, *n_segments* 2000, *compactness* 10 e *sigma* 3. A classificação automática representada na figura Figura 4.7(l) (Muro Escola Castelejo) foi produzida pelo classificador SVC, treinado com 27 imagens e com os seguintes parâmetros SLIC: *threshold* 0.5, *n_segments* 500, *compactness* 20 e *sigma* 1.

No primeiro grupo de 3 figuras (Antártida) verifica-se que se obtém uma segmentação perfeita. No entanto é importante referir que para exactamente as mesmas condições, mas considerando uma imagem diferente (Figura 4.7(c)), esse mesmo classificador produz apenas uma imagem completamente preta. Isto é um exemplo de falso negativo.

Relembrando a Figura 4.6(e) e Figura 4.6(g) referentes ao conjunto de Cascais, verifica-se que, independentemente do classificador e dos parâmetro SLIC, as segmentações são todas, com raras excepções, de baixa qualidade. De facto na Figura 4.7(i) observam-se vários falsos positivos.

No caso do exemplo representado na Figura 4.7(l) a classificação conseguiu até ser melhor que a classificação manual do GrabCut.

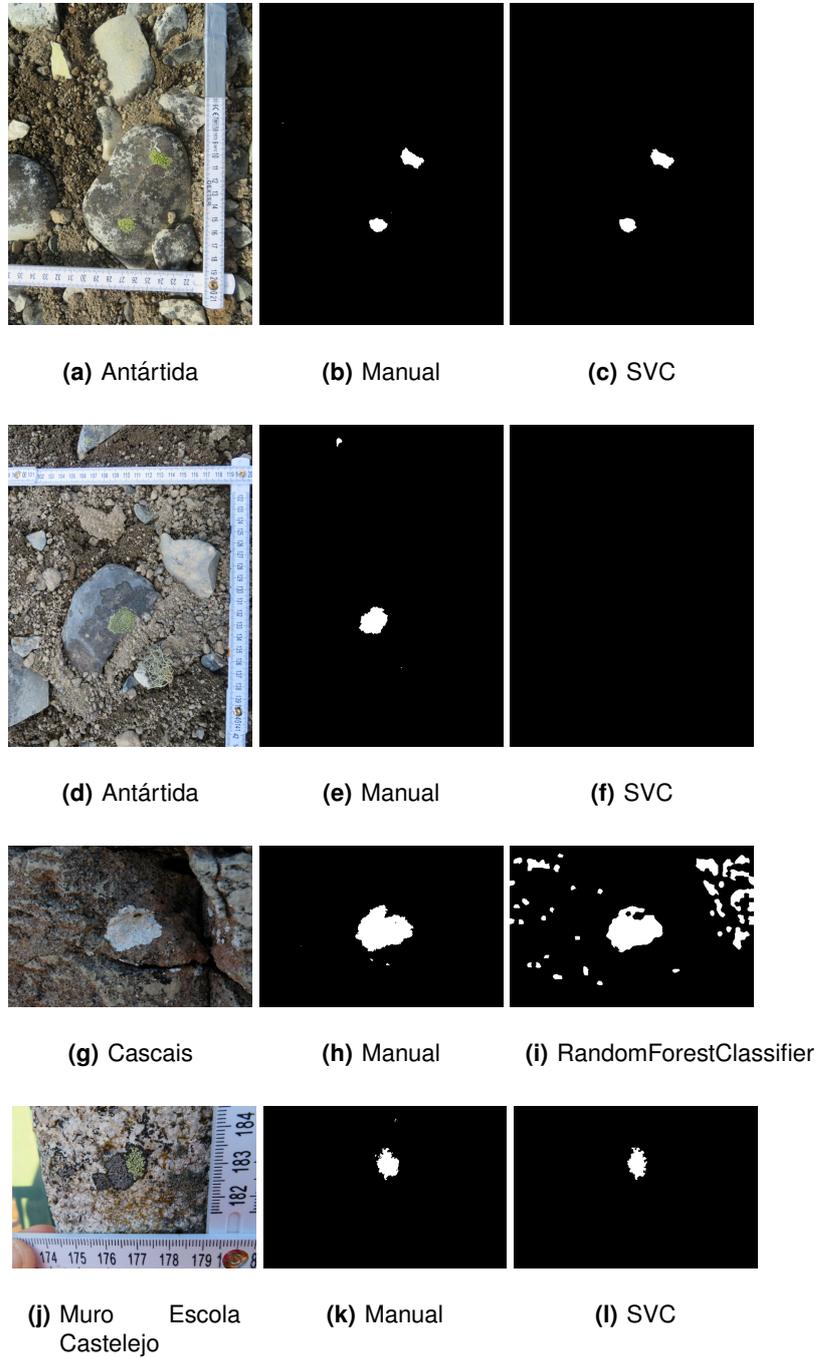


Figura 4.7: Exemplos de classificações automáticas de imagens seleccionadas dos conjuntos da Antártica, Cascais e Muro Escola Castelejo.

5

Discussão

Este trabalho focou-se em 3 componentes principais que incluem: (1) colocação e detecção das miras para atribuição de escala; (2) classificação manual de imagens a ser aplicada aos conjuntos de treino e de teste; (3) classificação automática de imagens. Neste capítulo, estas diferentes componentes serão discutidas, tendo em mente os objectivos iniciais deste trabalho (análise de imagem para medição automática de líquenes para liquenometria), desafios encontrados e melhor procedimentos para maximizar a qualidade dos resultados.

No que diz respeito à detecção das miras, e dado que a mesma é efectuada por segmentação de cor, observou-se que, em certos casos, o sistema apresenta alguns problemas, especificamente, quando a fotografia apanha o céu.

No entanto, este problema pode ser evitado de várias formas: (1) usando outra cor da mira; (2) fazendo uma calibração com base na cor; (3) ou de uma forma mais simples, fazer a amostragem fotográfica de forma a não incluir o céu.

Outra questão prende-se com a existência de superfícies irregulares. Nestes casos, a colocação das miras poderá não estar alinhada com a superfície rochosa colonizada por líquenes. Assim, os resultados poderão acumular erros, pois o processamento das imagens é bidimensional, sendo que os objectos têm profundidade. No entanto, estes desafios são igualmente partilhados na definição de escalas com recurso a métodos manuais (fotografias com escalas/réguas e georreferenciação com recurso a plataformas de sistemas de informação geográfica). Assim sendo, a solução apresentada tem vantagens pois permite automatizar o processo de correcção e atribuição de escala.

No que se refere à classificação manual, e comparando com a vectorização ou selecção assistida com programas de processamento de imagem, observou-se que se obtiveram ganhos significativos no tempo de processamento. Para além disso, verificou-se que a qualidade das classificações obtidas com o GrabCut está a par da qualidade obtida com as classificações obtidas com os outros métodos.

Quanto à classificação automática, optou-se por se deixar a definição dos parâmetros SLIC em aberto, de forma que os mesmos possam ser definidos pelo utilizador. Na verdade, é necessário ter em conta que este programa foi desenvolvido para funcionar em conjuntos de imagens desconhecidos. Dado que o parâmetro *n_segments* é bastante sensível à percentagem de cobertura de líquen nas imagens, que é à partida desconhecido, decidiu-se que esta abordagem permitirá gerar os melhores resultados de classificação. É de referir que esta percentagem poderá ser divergente dentro do mesmo conjunto de imagens que estarão sujeitas aos mesmo parâmetros SLIC. Nesse caso, e de forma a garantir a melhor classificação possível, podem-se seguir várias abordagens: (1) avaliar quais os melhores parâmetros gerais antes da segmentação final, com base num conjunto de teste que inclua a total amplitude percentagem de líquenes, mesmo avaliada visualmente de forma qualitativa; (2) efectuar a amostragem tendo em conta estes desafios, evitando criar diferenças de percentagem líquen/rocha devido à captura de imagens a diferentes distâncias das superfícies, ou criar conjuntos diferentes para

processamento, agrupados por distância/percentagem de cobertura.

Em relação aos resultados das segmentações verificou-se que a qualidade varia bastante, dependendo do conjunto de imagens em estudo. Para alguns conjuntos testados, todos os 24 modelos apresentam resultados semelhantes, mas tal pode não se verificar para outros conjuntos. De forma a compensar esta Incerteza, propõe-se a elaboração de conjuntos de imagens mais estandardizados. É importante referir que os melhores procedimentos definidos já aquando discussão do sistema das miras também contribui para esse objectivo.

No futuro se o trabalho aqui desenvolvido for aplicado, uma melhoria que se poderá efectuar ao sistema será utilizar a métrica **MC! (MC!)C** como função regente da validação cruzada, já que, como explicado nas secções anteriores, as métricas convencionais não são ideais. Esta métrica pode também ser utilizada para definir o número mínimo de imagens de treino. As curvas de desempenho produzidas para os casos de teste utilizados neste trabalho, mostram uma estabilização do MCC a partir das 20-25 imagens de treino, para os classificadores *Random Forest* e *SVC* (os que deram melhores resultados). Assim, e para garantir um processamento eficiente com baixo custo/benefício, propõe-se a utilização de 20-25 imagens de treino.

É importante referir que o programa apenas classifica uma espécie de líquen por conjunto. Isto quer dizer, que para cada ciclo de execução, as imagens binárias classificadas manualmente para efectuar o treino, apenas devem referir-se a uma única espécie de líquen. Se um dado conjunto possuir mais do que uma espécie de líquen e se, se pretender fazer a segmentação de cada uma delas, então será necessário executar o programa várias vezes, com conjunto de classificações binárias manuais diferentes para cada espécie. No entanto, em trabalho futuro de desenvolvimento do programa, facilmente se poderá incluir segmentação multi-classe.

A metodologia desenvolvida apresenta uma grande vantagem face às soluções baseadas em *deep-learning*, pois como foi constatado na análise dos resultados na Seção 4.4.1, não são necessários muitos dados para os classificadores automáticos convergirem. Este facto permite utilizar este sistema com sucesso, mesmo quando o número de dados é reduzido.

Mesmo assim, foram detectados alguns casos em que o sistema foi menos funcional, que incluem situações com tonalidades rocha/píxel muito próximas e difíceis de distinguir, correspondendo a um ponto fraco e a uma consequência inevitável da segmentação por cor. Na verdade, estas incertezas de delimitação de rocha/líquén permanecem quando é utilizada delimitação manual, sendo que existe uma contribuição de variância de decisões associada ao operador que será eliminada com esta automatização.

De modo a mitigar essas situações, para estudos futuros, propõe-se explorar soluções com base em imagens multiespectrais (visíveis e não só), conjuntamente com a utilização de índices vegetativos para auxiliar/facilitar a segmentação. No entanto, o sistema funciona bastante bem para conjuntos de

imagens RGB com bons contrastes entre líquenes e o fundo, bem fotografados (focados) com % de cobertura de líquenes semelhantes e tonalidades consistentes em cada imagem.

6

Conclusão

Considera-se que os objectivos propostos inicialmente foram alcançados, tendo sido produzido um conjunto de ferramentas que irão auxiliar significativamente a amostragem e processamento de dados para liquenometria. Este conjunto de ferramentas inclui a automatização das aquisições com miras, a ferramenta de classificação manual e, por fim, a ferramenta de classificação automática. Todas juntas, ou mesmo em separado, estas ferramentas melhoram e aceleram o processo de classificação e quantificação de amostras.

Os resultados são promissores. No que se refere ao processamento manual das imagens, obtêm-se ganhos superiores a 75% com precisão superior a 95%. O uso da classificação automática, consegue atingir precisão média superior a 70%. Os resultados de trabalho permitem uma agilização de processos até aqui muito lentos, tendo também grande potencial de automatização com precisão elevada.

O simples facto de se obterem automaticamente estimativas das áreas das coberturas dos líquenes nas rochas, já apresenta um benefício considerável. Estes benefícios incluem, quer a menor exigência de tempo dispensado por parte do utilizador, quer a menor variância nos resultados herdada das decisões do mesmo. É de referir que esta ferramenta irá ser particularmente útil no processamento de um elevado número de dados (desejável quando aplicado a datações), por possibilitar análises estatísticas mais robustas, de modo a contribuir para estudos que elaborem modelos de crescimento dos líquenes das zonas estudadas e aplicar a datação por liquenometria.

Bibliografia

- [1] M. A. Oliveira, "Boulder deposits related to extreme marine events in the western coast of Portugal." Ph.D. dissertation, Universidade de Lisboa, Portugal, 2017.
- [2] 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.
- [3] M. A. Oliveira, E. Llop, C. Andrade, C. Branquinho, R. Goble, S. Queiroz, M. C. Freitas, and P. Pinho (*in press*), "Estimating the age and mechanism of boulder transport related with extreme waves using lichenometry," *Progress in Physical Geography*, 2020.
- [4] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary region segmentation of objects in n-d images," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1, 2001, pp. 105–112 vol.1.
- [5] "3.1. cross-validation: evaluating estimator performance," 2020. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html
- [6] 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.
- [7] 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. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169555X21001446>
- [8] W. Locke, J. Andrews, and P. Webber, "A manual for lichenometry. british geomorphological research group," *Technical Bulletin*, vol. 26, pp. 1–47, 01 1979.
- [9] D. P. McCarthy and K. Zaniewski, "Digital analysis of lichen cover: A technique for use in lichenometry and lichenology," *Arctic, Antarctic, and Alpine Research*, vol. 33, no. 1, pp. 107–113, 2001. [Online]. Available: <https://doi.org/10.1080/15230430.2001.12003411>

- [10] C. Roux and J. Egea, "L'opographetum durieui egea et roux ass. nov., une association lichénique saxicole-calcicole, halophile." *Cryptogamie, Bryologie, lichénologie*, vol. 13, pp. 105–115, 1992.
- [11] J. L. Innes, "Lichenometry," *Progress in Physical Geography: Earth and Environment*, vol. 9, no. 2, pp. 187–254, 1985. [Online]. Available: <https://doi.org/10.1177/030913338500900202>
- [12] 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.
- [13] A. Seminara, J. Fritz, M. Brenner, and A. Pringle, "A universal growth limit for circular lichens," *Journal of The Royal Society Interface*, vol. 15, p. 20180063, 06 2018.
- [14] 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. [Online]. Available: <https://doi.org/10.1080/00291957408551968>
- [15] 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. [Online]. Available: <http://www.jstor.org/stable/40510489>
- [16] 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. [Online]. Available: [https://doi.org/10.1657/1523-0430\(2003\)035\[0203:ELABDA\]2.0.CO;2](https://doi.org/10.1657/1523-0430(2003)035[0203:ELABDA]2.0.CO;2)
- [17] 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. [Online]. Available: <https://doi.org/10.1111/j.1468-0459.2011.00004.x>
- [18] H. E. Trenbirth, "Lichenometry." *Geomorphological Techniques (Online Edition)*. British Society for Geomorphology; London, UK. ISSN: 2047-0371., 2010, ch. Section 4.2.7. [Online]. Available: https://www.geomorphology.org.uk/sites/default/files/geom.tech_chapters/4.2.7_Lichenometry.pdf
- [19] E. M. Claudia Gazzano, Sergio E. Favero-Longo and R. Piervittori, "Image analysis for measuring lichen colonization on and within stonework," *The Lichenologist, British Lichen Society*, vol. 41, no. 3, p. 299–313, May 2009.
- [20] R. Armstrong, "Lichens, lichenometry and global warming," pp. 32–35, Sep. 2004.
- [21] J. S. Lalley and H. A. Viles, "Recovery of lichen-dominated soil crusts in a hyper-arid desert," *Biodiversity and Conservation*, vol. 17, no. 1, pp. 1–20, Jan 2008. [Online]. Available: <https://doi.org/10.1007/s10531-007-9153-y>

- [22] L. G. Sancho, A. Pintado, and T. G. A. Green, “Antarctic studies show lichens to be excellent biomonitors of climate change,” *Diversity*, vol. 11, no. 3, 2019. [Online]. Available: <https://www.mdpi.com/1424-2818/11/3/42>
- [23] H. C. Serrano, M. A. Oliveira, C. Barros, A. S. Augusto, M. J. Pereira, P. Pinho, and C. Branquinho, “Measuring and mapping the effectiveness of the european air quality directive in reducing n and s deposition at the ecosystem level,” *Science of The Total Environment*, vol. 647, pp. 1531–1538, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0048969718330262>
- [24] P. Pinho, C. Barros, S. Augusto, M. Pereira, C. Máguas, and C. Branquinho, “Using nitrogen concentration and isotopic composition in lichens to spatially assess the relative contribution of atmospheric nitrogen sources in complex landscapes,” *Environmental Pollution*, vol. 230, pp. 632–638, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0269749116320371>
- [25] R. Armstrong, *The influence of environmental factors on the growth of lichens in the field*. Germany: Springer, 2015, vol. 1, pp. 1–18.
- [26] J. Orwin, K. MCKINZEY, M. STEPHENS, and A. Dugmore, “Identifying moraine surfaces with similar histories using lichen size distributions and the u2 statistic, southeast iceland,” *Geografiska Annaler: Series A, Physical Geography*, vol. 90, pp. 151 – 164, 06 2008.
- [27] W. Bull, “Using earthquakes to assess lichen growth rates,” *Geografiska Annaler: Series A, Physical Geography*, vol. 96, 06 2014.
- [28] “Photoshop User Guide,” 2021. [Online]. Available: <https://helpx.adobe.com/photoshop/using/making-quick-selections.html>
- [29] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. Berlin, Heidelberg: Springer-Verlag, 2010.
- [30] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2008.
- [31] “1. Supervised learning,” 2020. [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>

- [33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
- [35] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges," *The International Journal of Advanced Manufacturing Technology*, 05 2021.
- [36] C. Rother, V. Kolmogorov, and A. Blake, "'grabcut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, p. 309–314, Aug. 2004. [Online]. Available: <https://doi.org/10.1145/1015706.1015720>
- [37] "1.4 Support Vector Machines, scikit-learn 0.20.2 documentation," 2020. [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>
- [38] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "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.
- [39] W. Press, S. Teukolsky, W. H. Vetterling, B. Flannery, and S. A. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007, vol. Section 16.5. Support Vector Machines. [Online]. Available: <https://books.google.pt/books?id=1aAOdzK3FegC>
- [40] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug 1996. [Online]. Available: <https://doi.org/10.1007/BF00058655>
- [41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "Slic superpixels," *Technical report, EPFL*, 06 2010.
- [42] 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. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0005279575901099>
- [43] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS ONE*, vol. 12, no. 6, p. e0177678, Jun. 2017.
- [44] D. M. W. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

- [45] M. Kırıcı, E. O. Günes, Y. Çakır, and S. Şentiirk, "Vegetation measurement using image processing methods," in *2014 The Third International Conference on Agro-Geoinformatics*, 2014, pp. 1–5.
- [46] J. Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," *SpringerPlus*, vol. 2, p. 660, 12 2013.
- [47] Wei-Tai Chen, Yu-Hui Flora Yeh, Ting-Yu Liu, Ta-Te Lin, "An Automatic Plant Growth Measurement System for Plant Factory."
- [48] H. Shimizu and R. D. Heins, "Computer-Vision-Based System For Plant Growth Analysis," *Transactions of the ASAE*, vol. 38, no. 3, pp. 959–964, May 1995. [Online]. Available: <https://www.researchgate.net/publication/270613999>
- [49] W. J. C. J. Lin Kaiyan and S. Huiping, "Measurement of plant leaf area based on computer vision," in *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*, 2014.
- [50] M. Hassanein, Z. Lari, and N. El-Sheimy, "A new vegetation segmentation approach for cropped fields based on threshold detection from hue histograms," *Sensors*, vol. 18, p. 1253, 04 2018.
- [51] I. k. Nti, E. Gyamfi, and Y. Samuel, "Detection of plant leaf disease employing image processing and gaussian smoothing approach," *International Journal of Computer Applications*, vol. 162, pp. 20–25, 03 2017.
- [52] 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.
- [53] S. Salehi, M. Karami, and R. Fensholt, "Identification of a robust lichen index for the deconvolution of lichen and rock mixtures using pattern search algorithm (case study: Greenland)," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 973–979, 06 2016.
- [54] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.



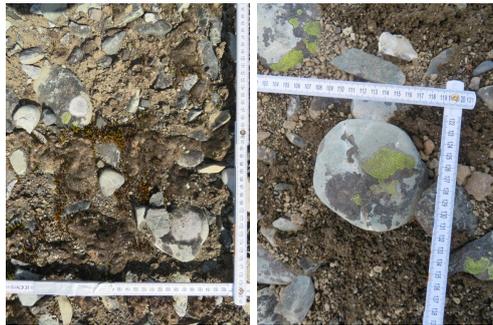
Anexos

Conjunto de imagens	nº do teste	classificador	tipo de características	modelo de cor	tempo de treino (seg.)	Precisão no conjunto de treino	Precisão no conjunto de teste	
dataset 1 small	1	RandomForestClassifier	SLIC	hsv	8,19	99,99%	96,16%	
		SVC	SLIC	hsv	12,69	92,15%	95,62%	
		RandomForestClassifier	SLIC	rgb	9,66	99,99%	95,54%	
	2	SVC	SLIC	rgb	17,67	99,99%	95,29%	
			RandomForestClassifier	pixel	422,48	99,99%	94,37%	
			SVC	SLIC	hsv	14,62	95,48%	92,89%
		GaussianNB	SLIC	hsv	0,18	94,31%	92,60%	
		RandomForestClassifier	nn_layer	rgb	1096,83	99,99%	90,41%	
		RandomForestClassifier	pixel	hsv	506,87	99,99%	90,05%	
		RandomForestClassifier	pixel	hsv	522,57	99,99%	89,73%	
		SVC	SLIC	hsv	16,34	99,99%	91,72%	
		RandomForestClassifier	SLIC	hsv	9,82	99,99%	91,42%	
dataset 2 small	3	RandomForestClassifier	nn_layer	rgb	982,22	99,99%	91,04%	
		SVC	SLIC	hsv	16,63	99,99%	90,14%	
		GaussianNB	SLIC	hsv	0,23	96,55%	89,85%	
	4	SVC	SLIC	rgb	14,94	99,60%	92,40%	
			GaussianNB	SLIC	rgb	0,22	98,40%	92,13%
			RandomForestClassifier	SLIC	rgb	9,04	99,99%	92,09%
		RandomForestClassifier	SLIC	hsv	7,52	99,99%	91,32%	
		RandomForestClassifier	pixel	hsv	493,05	99,99%	90,34%	
		RandomForestClassifier	SLIC	hsv	5,73	99,99%	95,89%	
		SVC	SLIC	hsv	8,67	96,23%	95,56%	
		RandomForestClassifier	pixel	hsv	535,99	99,99%	95,46%	
		RandomForestClassifier	nn_layer	rgb	1154,13	99,99%	94,81%	
dataset 3 small	5	RandomForestClassifier	pixel	593,98	99,99%	94,76%		
		GaussianNB	SLIC	rgb	0,19	74,15%	91,81%	
		RandomForestClassifier	SLIC	rgb	7,90	99,99%	89,89%	
	6	SVC	SLIC	rgb	61,56	93,83%	89,36%	
			RandomForestClassifier	pixel	767,91	99,99%	87,36%	
			RandomForestClassifier	pixel	762,09	99,99%	86,12%	
		RandomForestClassifier	nn_layer	hsv	2285,34	99,99%	99,89%	
		SVC	pixel	hsv	1272,09	99,84%	99,82%	
		SVC	SLIC	hsv	3,74	99,99%	99,76%	
		RandomForestClassifier	pixel	hsv	240,34	99,95%	99,76%	
		SVC	nn_layer	rgb	4878,10	99,88%	99,71%	
		SVC	nn_layer	rgb	4435,36	99,94%	99,57%	
dataset 3 small	7	RandomForestClassifier	nn_layer	hsv	2226,99	99,99%	99,46%	
		SVC	pixel	949,27	99,89%	99,44%		
		SVC	pixel	1017,76	99,89%	99,41%		
	8	DecisionTreeClassifier	nn_layer	rgb	102,82	99,99%	99,33%	
			nn_layer	rgb	4968,45	99,83%	99,42%	
			DecisionTreeClassifier	nn_layer	hsv	115,91	99,99%	99,29%
		RandomForestClassifier	nn_layer	hsv	2355,74	99,99%	99,25%	
		SVC	pixel	hsv	1128,82	99,56%	99,15%	
		RandomForestClassifier	pixel	hsv	218,80	99,90%	99,12%	

Tabela A.1: Tabela com os resultados dos melhores cinco pares classificadores-características dos nove testes para seleção e avaliação do modelo e características.

Conjunto de imagens	nº do teste	classificador	slc threshold	slc segments	slc compact	slc sigma	Melhores hiperparâmetros pelo Gridsearch	Média das taxas de positivos e negativos verdadeiros nos conjuntos de teste
1	1 small	RandomForestClassifier	0	500	10	3	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9550
		RandomForestClassifier	0	500	20	1	{criterion: 'gini', h.estimators: 150, random.state: 42}	0.9540
		RandomForestClassifier	0	500	10	1	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9518
		RandomForestClassifier	0	1000	20	3	{criterion: 'entropy', h.estimators: 100, random.state: 42}	0.9503
		RandomForestClassifier	0.5	2000	10	1	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9488
		RandomForestClassifier	0.5	2000	20	3	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9152
		SVC	0.5	500	20	1	{C: 10, gamma: 'scale', kernel: 'rbf', max.iter: 500}	0.9089
		RandomForestClassifier	0.5	500	20	3	{criterion: 'gini', h.estimators: 150, random.state: 42}	0.9073
		RandomForestClassifier	0.5	1000	20	3	{criterion: 'gini', h.estimators: 100, random.state: 42}	0.9058
		SVC	0.5	500	20	3	{C: 10, degree: 2, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9354
		SVC	0.5	500	10	3	{C: 10, degree: 4, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9331
		SVC	0.5	1000	20	1	{C: 10, degree: 2, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9294
2	2 small	SVC	0.5	2000	10	1	{C: 10, degree: 2, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9277
		SVC	0	500	20	3	{C: 1, degree: 2, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9410
		SVC	0	500	20	3	{C: 1, degree: 3, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9389
		SVC	0	1000	20	3	{C: 1, degree: 3, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9322
		SVC	0.5	500	20	3	{C: 1, degree: 2, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9315
		SVC	0.5	2000	20	3	{C: 100, degree: 5, gamma: 'auto', kernel: 'poly', max.iter: 500}	0.9302
		RandomForestClassifier	0	2000	20	1	{criterion: 'entropy', h.estimators: 100, random.state: 42}	0.9519
		RandomForestClassifier	0	2000	20	3	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9499
		SVC	0	2000	20	3	{C: 1, gamma: 'scale', kernel: 'rbf', max.iter: 1000}	0.9496
		SVC	0	1000	20	1	{C: 1, gamma: 'scale', kernel: 'rbf', max.iter: 500}	0.9495
		SVC	0	2000	10	3	{C: 1, gamma: 'scale', kernel: 'rbf', max.iter: 1000}	0.9471
		3	3 small	RandomForestClassifier	0.5	1000	20	3
RandomForestClassifier	0.5			2000	20	1	{criterion: 'gini', h.estimators: 100, random.state: 42}	0.9513
RandomForestClassifier	0.5			1000	20	3	{criterion: 'entropy', h.estimators: 100, random.state: 42}	0.9495
RandomForestClassifier	0.5			1000	10	3	{criterion: 'entropy', h.estimators: 100, random.state: 42}	0.9470
RandomForestClassifier	0.5			500	20	1	{criterion: 'gini', h.estimators: 100, random.state: 42}	0.9470
RandomForestClassifier	0			2000	10	3	{criterion: 'gini', h.estimators: 150, random.state: 42}	0.9737
RandomForestClassifier	0			2000	20	3	{criterion: 'gini', h.estimators: 100, random.state: 42}	0.9708
RandomForestClassifier	0			500	20	3	{criterion: 'gini', h.estimators: 150, random.state: 42}	0.9631
RandomForestClassifier	0			1000	10	3	{criterion: 'gini', h.estimators: 100, random.state: 42}	0.9599
RandomForestClassifier	0			500	20	1	{criterion: 'entropy', h.estimators: 150, random.state: 42}	0.9598
SVC	0			2000	10	3	{C: 100, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.7934
4	4 small			SVC	0.5	2000	10	3
		SVC	0.5	2000	20	3	{C: 10, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.7288
		SVC	0.5	2000	10	3	{C: 100, degree: 3, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.7252
		SVC	0.5	2000	10	3	{C: 100, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.7243
		RandomForestClassifier	0	500	10	3	{criterion: 'entropy', h.estimators: 50, random.state: 42}	0.9572
		SVC	0	2000	20	1	{C: 10, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9195
		SVC	0	2000	20	3	{C: 10, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.9136
		SVC	0	2000	10	3	{C: 10, gamma: 'scale', kernel: 'poly', max.iter: 1000}	0.8531
		SVC	0	1000	10	3	{C: 10, gamma: 'scale', kernel: 'poly', max.iter: 500}	0.8518

Tabela A.2: Tabela com os resultados dos melhores cinco pares classificadores-características SLIC dos nove testes para selecção e avaliação dos modelos e características SLIC.



(a) Antártida

(b) Antártida

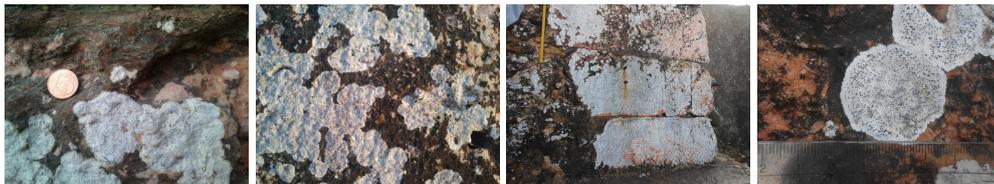


(c) Muro Escola Castelejo

(d) Muro Escola Castelejo

(e) Muro Castelejo 2

(f) Muro Castelejo 2



(g) Nazaré 1

(h) Nazaré 1

(i) Nazaré 2

(j) Nazaré 2



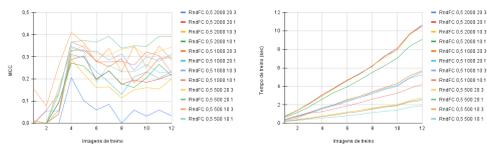
(k) Nazaré 3

(l) Nazaré 3

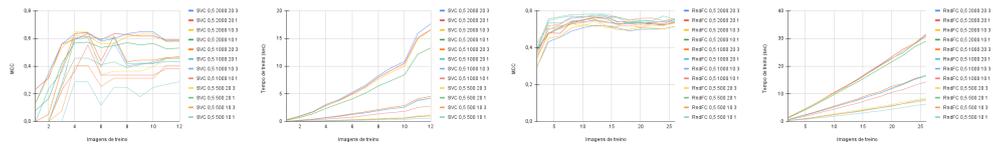
(m) Cascais

(n) Cascais

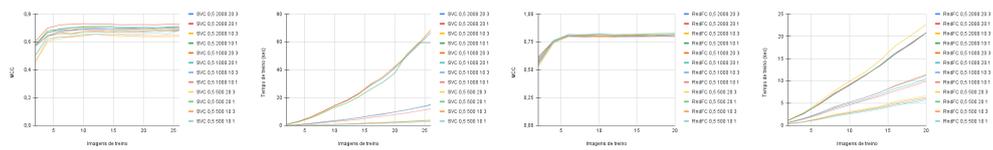
Figura A.1: Exemplos de duas imagens de cada conjunto de dados.



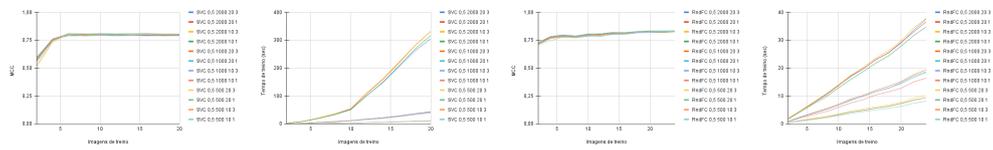
(a) Muro Escola Castelejo **(b) Muro Escola Castelejo**



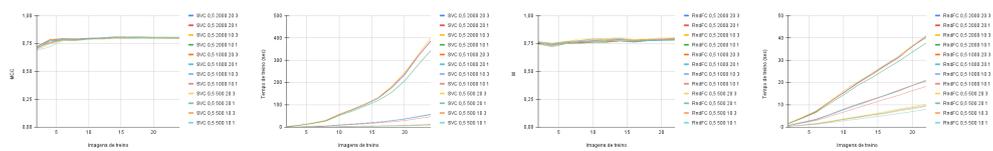
(c) Muro Escola Castelejo 2 **(d) Muro Escola Castelejo 2** **(e) Muro Castelejo** **(f) Muro Castelejo 2**



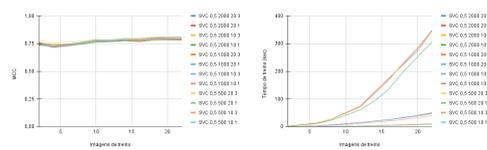
(g) Muro Castelejo 2 **(h) Muro Castelejo 2** **(i) Nazaré 1** **(j) Nazaré 1**



(k) Nazaré 1 **(l) Nazaré 1** **(m) Nazaré 2** **(n) Nazaré 2**



(o) Nazaré 2 **(p) Nazaré 2** **(q) Nazaré 3** **(r) Nazaré 3**



(s) Nazaré 3 **(t) Nazaré 3**

Figura A.2: Curvas dos restantes conjuntos de dados.