

# Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging

Dani Silva

*2013155412, Mestrado Integrado em Engenharia Biomédica*

Filipa Costa

*2013148125 Mestrado Integrado em Engenharia Biomédica*

---

## Abstract

Melanomas malignos, a maneira mais mortal do cancro da pele, tem um bom prognóstico se for tratado logo num estado inicial ainda curável. O diagnóstico precoce e a remoção do melanoma por cirurgia são os tratamentos mais eficazes do melanoma.

Este artigo apresenta as etapas computacionais do diagnóstico automatizado do melanoma. Estas etapas computacionais são: 1.) pré-processamento e segmentação de imagens para determinar as áreas fulcrais, 2.) extração e seleção de features para quantificar as propriedades, e 3.) classificar as áreas da lesão como malignas ou não. Neste artigo, detalhamos estas etapas computacionais e tentamos contornar os diversos desafios que o *data set* vai apresentando.

A melhor classificação final proporciona uma sensibilidade de 74% com uma especificidade de 75%.

Os resultados da classificação foram obtidos num conjunto de 764 imagens dermatoscópicas que pertencem ao arquivo do ISIC (*International Skin Imaging Collaboration*) fornecidas no Concurso *Skin Lesion Analysis Towards Melanoma Detections*.

---

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Estado de Arte</b>	<b>4</b>
2.1	Segmentação de imagens . . . . .	4
2.2	Extração de features . . . . .	6
2.3	Seleção de features . . . . .	6
2.4	Classificadores . . . . .	7
<b>3</b>	<b>Dataset</b>	<b>9</b>
<b>4</b>	<b>Pré-processamento e Segmentação</b>	<b>9</b>
4.1	Resultados . . . . .	14
<b>5</b>	<b>Extração de Features</b>	<b>16</b>
5.1	Features de forma . . . . .	17
5.2	Features de cor . . . . .	17
5.3	Features de textura . . . . .	18
<b>6</b>	<b>Seleção de Features</b>	<b>19</b>
<b>7</b>	<b>Classificação</b>	<b>22</b>
7.1	Classificadores convencionais . . . . .	22
7.2	Outros Classificadores . . . . .	23
7.3	Resultados . . . . .	26
<b>8</b>	<b>Discussão</b>	<b>27</b>
8.1	Segmentação . . . . .	27
8.2	Extração e Seleção de Features . . . . .	28
8.3	Classificação . . . . .	28
<b>9</b>	<b>Conclusão</b>	<b>29</b>

## 1. Introdução

Melanoma é o tipo mais grave de cancro de pele, desenvolve-se nas células (melanócitos) que produzem melanina - o pigmento que dá cor à pele. A causa exata de todos os melanomas não é clara, mas a exposição à radiação ultravioleta (UV) da luz solar ou lâmpadas de bronzamento aumenta o risco de desenvolver melanoma.

O risco de melanoma parece vir a aumentar a cada ano, sendo responsável por mais de 9.000 mortes por ano apenas nos Estados Unidos, e mais de 10.000 mortes estimadas em 2016 [1]. Conhecer os sinais de alerta deste tipo de cancro da pele pode ajudar a garantir que as alterações cancerígenas são detetadas e tratadas antes que o cancro se desenvolva.

Como o melanoma ocorre na superfície da pele, é passível de deteção por um simples exame visual. Na verdade, a maioria dos melanomas são reconhecidos pela primeira vez pelos pacientes, não pelos médicos[2]. No entanto, a inspeção visual sem auxílio de dermatologistas especializados está associada a uma precisão diagnóstica de cerca de 60%, o que significa que muitos melanomas curáveis não são detetados até estágios mais avançados. Para melhorar o desempenho do diagnóstico e reduzir as mortes por melanoma, foi introduzida a dermoscopia, que é uma técnica de imagem que elimina a reflexão da superfície da pele, permitindo que camadas mais profundas possam ser visualizadas com maior pormenor. Assumindo níveis adequados de perícia pelo intérprete, a imagem dermatoscópica mostrou melhorar o desempenho de reconhecimento de melanomas, resultando numa melhoria da precisão absoluta para valores de 75% -84% [3]. No entanto, tem sido demonstrado que a dermoscopia pode realmente diminuir a precisão diagnóstica nas mãos de dermatologistas inexperientes [4]. Além disso, para o diagnóstico, os dermatologistas contam com sua experiência clínica e percepção visual. No entanto, o diagnóstico feito pela visão humana é um tanto subjetivo, faltando precisão e reprodutibilidade.

Os sistemas de análise de imagens por dermoscopia computadorizada não têm a limitação dessa subjetividade. Estes sistemas permitem a utilização de um computador como um segundo método de diagnóstico independente, que pode ser potencialmente utilizado para a pré-selecção de doentes, realizados por operadores não experientes e para a assistência dos clínicos. Embora as técnicas de análise computadorizada não possam fornecer um diagnóstico definitivo, estas podem ser usadas para melhorar a tomada de decisão de biópsia da mancha, o que alguns observadores acham que é o uso mais im-

portante para a dermatoscopia [5]. Finalmente, a análise automatizada pode servir como uma ferramenta adicional para melhorar o acompanhamento.

À medida que o desenvolvimento em aplicações com base em imagens dermatoscópicas para smart phones estão a chegar ao mercado [6], aumenta também a investigação em algoritmos de avaliação dermoscópica para influenciar positivamente o cuidado do paciente. Como resultado, o interesse da comunidade tem crescido, como muitos centros de investigação a reunirem os seus esforços de pesquisa em análise automatizada.

Neste trabalho, é apresentada uma abordagem metodológica para a classificação de imagens de dermatoscopia, com base no banco de imagens da ISIC. O resto do artigo está organizado da seguinte forma: Seção 2 descreve as várias técnicas e metodologias utilizadas até à atualidade. Seção 3 descreve o conjunto de dados utilizado na análise. A seção 4 explica o pré-processamento, a segmentação efetuada e os resultados obtidos pelas mesmas. A Seção 5 discute a extração de features. A seção 6 descreve a seleção de features. A Seção 7 apresenta as várias famílias de classificação utilizadas e os resultados obtidos. A Seção 8 apresenta as discussões consequentes dos resultados obtidos. Finalmente a Seção 9 apresenta a conclusão e os trabalhos futuros.

## 2. Estado de Arte

A abordagem padrão na análise automática de imagens dermatoscópicas tem normalmente três fases: A) Segmentação da imagem; B) Extração de features; C) Seleção de features e D) Classificação.

### 2.1. Segmentação de imagens

A segmentação é a fase mais importante para uma análise correta de imagens, uma vez que vai influenciar a precisão das etapas subsequentes. No entanto, a fase de segmentação não é uma tarefa simples, devido à grande variedade de lesões, tipos de pele, presença de pelos [7]. Para resolver este problema, vários algoritmos têm sido propostos, podendo ser classificados como thresholding (o mais utilizado), edge-based ou region-based.

Em 1994, Green et al.[8] o algoritmo utilizado para a segmentação de imagens foi uma adaptação do thresholding para um tridimensional espaço de cor, obtendo 83,8% de segmentação correta (171 aproveitáveis imagens em 204 imagens de lesões).

Em 1998, Hance et al. [9] comparou 6 diferentes algoritmos de segmentação da cor (adaptive thresholding, fuzzy-c-means, SCT/center split [10], PCT/median, split and merge, and multiresolution segmentation). Numa gama de 66 imagens, os algoritmos que alcançaram uma melhor performance foram o adaptive thresholding (40 em 66) e o PCT/median (46 boas segmentações).

Em 2001, Ganster et al. [11] usaram 3 algoritmos diferentes para segmentar as imagens de melanomas, denominados por global thresholding, dynamic thresholding, e a 3-D color clustering concept. De um conjunto de 4000 imagens de lesões de pele, apenas 159 foram rejeitadas devido a erros de segmentação, nomeadamente quando a máscara não conseguia cobrir a lesão toda ou quando a diferença da cor entre a lesão e a pele saudável era mínima. Portanto, 96% das imagens das lesões foram corretamente segmentadas.

Em 2005, Erkol et al. [12] verificaram a qualidade do GVF-based method para a segmentação de 100 imagens de lesões da pele, usando outros 2 métodos para comparação - color histogram analysis technique e a segmentação manual por um segundo conceituado dermatologista. Os resultados obtidos foram baseados numa comparação com uma segmentação manual das lesões determinada por um primeiro dermatologista. A média dos erros obtidos pelo método GVF eram muito menores quer para as imagens benignas quer para as de melanomas (13.77% para benigno e 19.76% para melanoma) comparativamente aos erros obtidos pelo 2º método (19.87% e 91.96%). Ainda assim eram maiores do que os erros obtidos pelo segundo dermatologista (8,71% e 8,13%). Um dos maiores problemas do GVF aplicado a imagens dermatoscópicas são as bordas “fracas” em algumas das imagens.

Em 2009, Silveira et al. [13] avaliaram 6 métodos para a segmentação de lesões da pele em 100 imagens dermoscópicas: Adaptive thresholding (AT), Gradient Vector Flow (GVF), Adaptive Snake (AS), Level Set Method of Chan et al. (C-LS), Expectation-maximization level set (EM-LS) e fuzz-based split-and-merge algorithm (FBSM). Os resultados obtidos foram comparados com as imagens de “referência” (ground truth) e os erros de segmentação foram avaliados por 4 métodos diferentes de avaliação. O TDR com considerado dos 4 o mais relevante método de avaliação do ponto de vista da clinica, sendo que os melhores métodos de segmentação obtidos foram AS e EM-LS (que são semi-supervisionados, com TDR=95,47% e TDR=95,20% respetivamente) e o FBSM (que é um método completamente automático, com TDR=93,67%).

## 2.2. *Extração de features*

Existem inúmeros estudos e métodos que permitem extrair características relevantes e importantes de um determinado conjunto de funcionalidades, diminuindo a complexidade computacional dos algoritmos.

- Forma: é uma importante característica quando se trata de manchas na pele. A área da lesão deve ser obtida pelo método de quadriláteros de bits uma vez que, garante melhores resultados para objetos com bordas irregulares, o que não acontece por contagem do número de pixels dentro da borda [14]. É relevante também obter a proporção entre o comprimento do eixo maior e o comprimento do eixo menor; assimetria da lesão e a compactação da lesão [14] para desta forma adquirir resultados mais viáveis;
- Cores: Para quantificar as cores que a lesão apresenta pode-se utilizar a média e o desvio padrão, assimetrias severas na cor e histograma de distâncias [14];
- Textura: dependendo da técnica usada podem-se obter diversas características à cerca da textura da lesão. Um dos métodos que permite obter dezenas de dados estatísticos descritivos da textura é com base na matriz de co-ocorrência dos níveis de cinza (GLCM) [14].

Barata et al. [15] foi um pouco mais longe e analisou a importância da textura comparativamente às cores na classificação de imagens dermatológicas, concluindo pelos resultados obtidos que apenas os dados fornecidos pela cor da lesão são mais assertivos do que apenas dados sobre a textura ou mesmo ambos.

## 2.3. *Seleção de features*

A seleção das características é um importante passo de pré-processamento em muitas tarefas de machine learning. O objetivo é reduzir o número de características consideradas para o estudo, eliminando features redundantes, irrelevantes ou ruidosos. Do ponto de vista de classificação, existem inúmeros benefícios associados com a seleção de características: (i) tempo de armazenamento, (ii) a redução da complexidade do classificador, (iii) aumento da precisão, (iv) redução dos tempos de teste, e (v) uma melhor compreensão e visualização de dados.

Os métodos de filtragem dependem de características gerais dos dados para selecionar um subconjunto de recursos sem envolver qualquer algoritmo de aprendizagem.

Na literatura proposta para lesões de pele existem muitos métodos para fazer a seleção de features, de seguida apresentamos alguns com bom desempenho em conjuntos de diversos dados:

- ReliefF: Neste algoritmo, um número de amostras é selecionado aleatoriamente a partir do conjunto de dados inicial e são determinados os vizinhos mais próximos, podendo ter em conta o ruído e dados multi-classes [14].
- Mutual information based feature selection (MIFS): avalia a informação igual entre as características individuais e os nomes das classes, e seleciona as features que tenham o máximo de informação em comum [14].
- Correlation based feature selection (CFS): Este algoritmo tenta encontrar um conjunto de recursos que individualmente se correlacionam bem com a classe [14].
- O SVM pode ser utilizado para selecionar features pois apresenta uma ampla gama de aplicações e é um método simples que se provou ser robusto e preciso quando comparado com outras abordagens [16].

#### 2.4. *Classificadores*

A classificação é o processo de extração de informação em imagens para reconhecer padrões e objetos homogêneos sendo que os estudos relacionados com a classificação automática de imagens de lesões pigmentadas na pele têm aparecido na literatura desde 1987 [17]. Com base em imagens dermatoscópicas e após os processos de segmentação e extração de features podem ser aplicados diferentes tipos de classificadores:

1. Support vector machines (SVMs): é um dos classificadores que desperta mais atenção na comunidade de Machine Learning, devido à sua sólida formação teórica e excelente desempenho prático. SVMs têm várias vantagens sobre os classificadores mais clássicos, como árvores de decisão e redes neurais [14]. Como forma de melhorar o desempenho deste classificador houve recente uma pesquisa mais intensiva relativa ao desequilíbrio de classes na comunidade de Machine Learning. Esse

problema geralmente ocorre quando uma ou mais classes se sobrepõem a outras. Em tais casos, a maioria dos classificadores concentra-se na análise das grandes classes, e em consequência diminui a precisão da classificação das classes mais pequenas. Em muitos domínios, tais como diagnósticos médicos as consequências de uma classificação incorreta, tal como classificar as amostras minoritárias (melanoma) como amostras benignas, são bastante graves. De acordo com Celebi et al. [14] para evitar erros provocados por classes desequilibradas deve-se analisar a área sob a curva ROC (gráfico de True Positive versus false positive). Com base na pesquisa de Celebi et al. [14] a abordagem efetuada utilizando SVM obteve ótimos resultados (especificidade de 92.34% e sensibilidade de 93.33%). Sendo que é necessário realçar que no número de features usadas na implementação do algoritmo ( $k$ ) é um parâmetro importante que deve ser considerado de modo a obter-se o melhor desempenho possível do classificador.

2. KNN: Pela análise da investigação realizada por Barata [15], verifica-se que utilizando o classificador KNN obtêm-se bons resultados de especificidade e sensibilidade, fazendo variar o número de vizinhos ( $k$ ) e a distância ( $SE = 100\%$  e  $SP = 72\%$ ). De forma a garantir uma boa generalização do método comparou-se o KNN ao algoritmo SVM verificando-se que este apresenta uma performance inferior ( $SE = 92\%$  e  $SP = 72\%$ ).
3. AdBoost: tem a capacidade de selecionar os recursos apropriados para um determinado problema e elimina features que não melhoram a performance do classificador. Para melhorar o desempenho deste tipo de classificador é necessário também ter em conta o número de bins pois influenciará o número de features a considerar. Assim de acordo com o procedimento efetuado por Barata [15] alcançam-se valores de  $SE = 92\%$  e  $SP = 85\%$ .
4. C4.5: O algoritmo C4.5 constrói árvores de decisão a partir de um conjunto de dados de treino, utilizando o conceito de entropia. Em cada nó da árvore, o algoritmo C4.5 escolhe o atributo dos dados que mais particiona o conjunto de amostras em subconjuntos tendendo a uma categoria ou a outra. O critério de particionamento é o ganho de informação normalizado (diferença de entropia). Neste tipo de classificador é necessário definir corretamente o número mínimo de samples per leaf ( $M$ ) e o fator de confiança [18]. O desempenho do algoritmo C4.5 na investigação de Celebi [18] foi então de 84,33% de sensibilidade



e especificidade de 96,19%.

5. Artificial Neural Networks (ANN): são inspiradas pelo sistema nervoso central de um animal que são capazes de realizar uma aprendizagem a partir inputs e seguidamente reconhecer padrões entre eles para tomar uma decisão global, requerem um treino inicial para identificar o conjunto de parâmetros da rede de acordo com os referidos pesos [19]. James et al. [20], tal como outros métodos referidos anteriormente, para testar a validade deste classificador obteve a curva ROC, determinando a probabilidade de uma lesão de pele ser classificada como melanoma corretamente. Os resultados obtidos foram os seguintes: sensibilidade, 94,3% e com um valor semelhante de especificidade 93,8%. Rubegni et al. [21] também usou artificial neural network como classificador e obteve valores de SE = 96% e SP=93%.

### 3. Dataset

O dataset usado neste projeto consiste num conjunto de 3 grupos de imagens dermoscópicas de lesões na pele (melanomas ou não melanomas) que pertencem ao arquivo do ISIC (International Skin Imaging Collaboration) fornecidas no Concurso Skin Lesion Analysis Towards Melanoma Detection. Essas imagens foram recolhidas dos principais centros clínicos a nível internacional e adquiridas a partir de uma variedade de dispositivos dentro de cada centro.

1. *Training data*: conjunto de 900 em formato JPEG. Todas as imagens usam o esquema de nome ISIC\_<image>\_id.jpg, onde <image>\_id é a identificação única de 7 dígitos. Estas imagens apresentam uma resolução espacial variável;
2. *Data*: conjunto de 379 imagens para testar o classificador;
3. *Ground Truth*: conjunto de imagens geradas por um grupo de especialistas em melanomas, utilizadas para comparação das imagens obtidas em cada fase do concurso. Estas imagens estão em forma de máscaras binárias.

### 4. Pré-processamento e Segmentação

No âmbito do concurso proposto pelo professor fomos convidados, numa primeira fase a apresentar previsões automáticas de segmentações de lesão a

partir das imagens dermoscópicas do dataset. Nesta etapa foi-nos fornecido um banco de imagens (ground truth) realizado manualmente por especialista e que utilizamos como padrão de comparação.

A etapa da segmentação e do pré-processamento são das mais importantes, uma vez que afeta a precisão dos passos subsequentes. No entanto, a segmentação é difícil por causa da grande variedade de formas de lesão, tamanhos (Fig. 1c) e cores (Fig. 1f), junto com diferentes tipos de pele e texturas. Além disso, algumas lesões têm limites irregulares (Fig. 1b) e, em alguns casos, há uma transição suave entre a lesão e a pele (Fig. 1e). Outras dificuldades estão relacionadas com a presença de objetos irrelevantes (Fig. 1d), presença de pelos sobre as lesões (Fig. 1a) e o mau enquadramento das mesmas na fotografia.

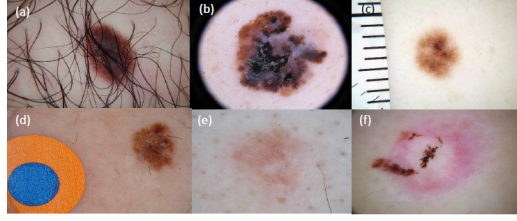


Figura 1: As figuras acima representam as mais diversificadas imagens do nosso *Dataset*. a) pelos, b) bordas pretas, c) réguas, d) objetos circulares, e) manchas muito ténues, f) Manchas claras com pequenas manchas escuras.

Todo o tratamento das imagens do dataset nesta fase foi realizado com recurso ao MATLAB®, incluindo diversas funções intrínsecas do mesmo.

Como se pode observar pela Fig. 2 foi necessário remover as regiões escuras nos quatro cantos das imagens. Para tentar resolver este problema, construiu-se um método que percorre a matriz dos pixels das imagens RGB. Caso os pixels que se encontram entre menos de  $1/4$  ou mais de  $3/4$  das margens da imagem corresponderem a uma cor muito escura (ou seja  $RGB < (40,40,40)$ ) estes são substituídos por valores correspondentes a uma cor semelhante à cor da pele ( $RGB=(255,184,131)$ ).

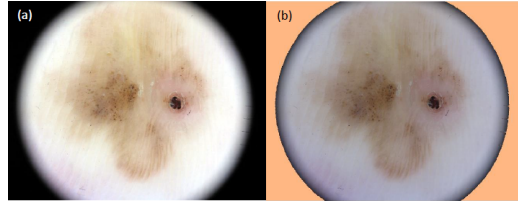


Figura 2: A figura (a) representa a imagem original de uma lesão. A figura b) representa a imagem em RGB após a implementação do código de alteração das cores das bordas da imagem.

Seguidamente, para destacar as manchas da pele efetuou-se um ajuste automático do contraste da imagem utilizando a função `imadjust`. Como se observa na Fig. 3, após a aplicação do ajuste a mancha torna-se mais saliente.

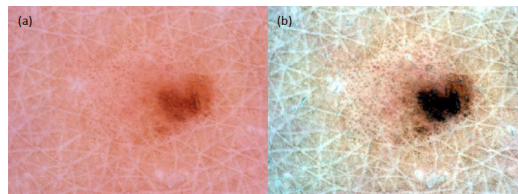


Figura 3: Comparação da imagem com aplicação da função `imadjust`. (a) imagem real; (b) imagem após aplicação do ajuste de contraste.

Para posteriormente ser possível aplicar a máscara binária é necessário eliminar a tridimensionalidade das matrizes associadas às imagens RGB e para tal procedeu-se à transformação das imagens para escala de cinza (matriz bidimensional). Deste modo resulta uma imagem binária e é facilitado o tratamento da mesma. Para além disso a escala de cinzentos na imagem permite diminuir a influência de objetos existentes inicialmente nas imagens e que induzem a erros de segmentação caso sejam mantidos, tais como os pelos e estruturas alongadas e escuras.

Para aplicar a escala de cinza a uma imagem RGB, foi efetuado um método que altera a coloração de forma bastante eficaz, tal como observado na Fig 4.

A partir da imagem em tons de cinza foi aplicado o algoritmo Otsu, em MATLAB, implementado pela função `multithresh` (Fig 5). Este método determina o valor ideal de um threshold que separe os elementos do fundo e da frente da imagem em dois clusters, atribuindo a cor branca ou preta para cada um deles (recorrendo à função `im2bw`).

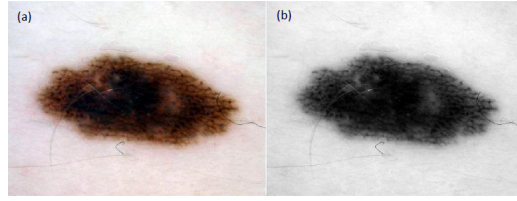


Figura 4: Comparação da imagem com a aplicação da escala de cinza. (a) imagem original; (b) imagem após aplicação do método.

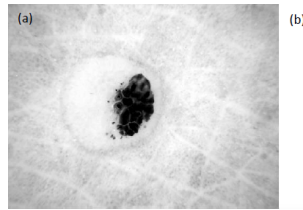


Figura 5: Comparação da imagem com a aplicação do thresholding e função im2bw. (a) imagem com aplicação de escala cinza; (b) imagem após aplicação da função multithresh e im2bw.

Partindo das imagens binárias foi então necessário retirar os pelos e outras estruturas irrelevantes (Fig 6). Para tal utilizaram-se as seguintes funções: strel, imrode e imdilate.

Utiliza-se imrode para “corroer” ligeiramente as bordas das estruturas na imagem em pequenas formas de disco (através da função strel) e desta forma uniformizam-se as margens. Seguidamente, aplica-se imdilate para recuperar o volume da mancha perdido após a aplicação da corrosão da imagem.

Após a comparação do dataset ground truth com as imagens binárias obtidas, concluiu-se que não correspondiam, logo foi necessário inverter as duas cores, o preto passa a branco e vice-versa. Para este ponto foi utilizada a função `incomplement` cujos resultados observam-se na Fig 7.

Devido à inconsistência de certas manchas foi necessário arranjar uma solução para que as mesmas se tornassem uniformes após tratamento. Para tal, inicialmente aplicou-se a função `imclose`, que une manchas que se encontram a determinada distância uma da outra. Seguidamente, para preencher os espaços no seu interior, como observado na Fig 8, utilizou-se a função `imfill(holes)`.

Por observação do banco de imagens em análise é possível verificar um número relativamente elevado de imagens com adesivos circulares (Fig 9a),

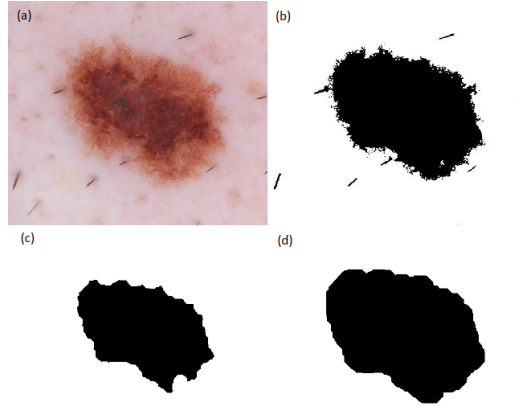


Figura 6: Comparação das imagens com implementação das funções strel, imrode e imdilate. (a) imagem original; (b) imagem tratada e após máscara binária; (c) imagem após aplicada a função imrode; (d) imagem com implementação da função imdilate.

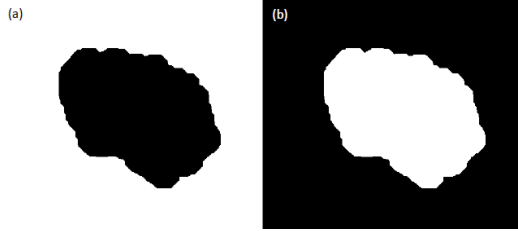


Figura 7: Comparação da imagem com a aplicação da função incomplement. (a) imagem tratada e após aplicação da máscara binária; (b) imagem após aplicação da função incomplement.

sem nenhuma relevância para a segmentação. Para tal foi necessário realizar um método eficaz para os eliminar. O procedimento passou por identificar todos os objetos presentes na imagem binária (função regionprops), através da localização, da forma e da área. Depois da identificação verificou-se os objetos quase lineares (estruturas irrelevantes) e os objetos perfeitamente circulares (adesivos) e eliminaram-se.

Após observação da Fig 10, deparamo-nos com outro problema que influencia de forma drástica os resultados. Certas imagens apresentam marcas nos cantos devido à luminosidade da mesma. Assim, foi necessário criar outro método que solucionasse este problema. A solução passou por verificar a presença de objetos com o seu centro nas margens da imagem, para tal foi considerado  $1/6$  da largura e do comprimento, como se verifica na Fig 10a.

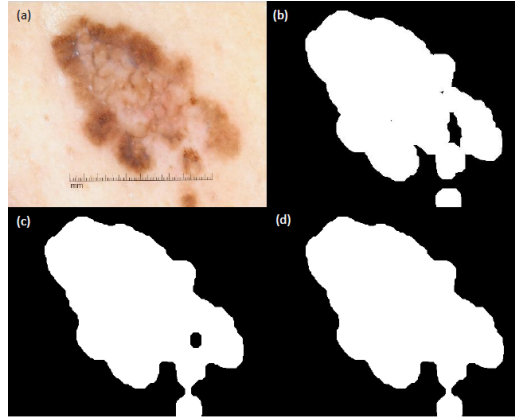


Figura 8: Comparação das imagens com implementação das funções imcomplement e imfill. (a) imagem original; (b) imagem tratada e após máscara binária; (c) imagem após aplicada a função imcomplement; (d) imagem após implementação da função imfill.

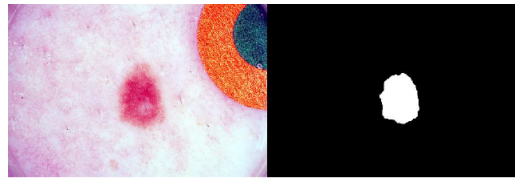


Figura 9: Comparação da imagem com a aplicação do método para retirar estruturas irrelevantes. (a) imagem com ajuste de contraste; (b) imagem após aplicação do método para eliminar adesivos circulares.

#### 4.1. Resultados

A verificação dos resultados da segmentação foi feita em comparação com o conjunto de dados do Ground Truth. As diferenças entre as imagens foram avaliadas por 4 métricas (pedidas no Concurso): Sensibilidade, Especificidade, Coeficiente de Dice e Jaccard (Tab. 1). A métrica que consideramos mais significativa foi a Jaccard, que tem um valor médio de 0,7206 (Fig. 11).

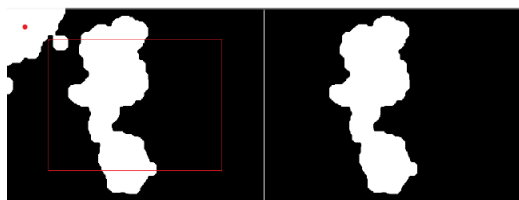


Figura 10: Verificação da utilização do método utilizado para remover objetos nas margens. (a) observação da área de análise dos objetos irrelevantes; (b) imagem após eliminação de estrutura irrelevante.

Tabela 1: Valores médios obtidos em cada método de avaliação para a segmentação.

	Média	Desvio Padrão
<b>Sensibilidade</b>	0,8151	0,1990
<b>Especificidade</b>	0,9532	0,1111
<b>Coefficiente de Dice</b>	0,8165	0,1813
<b>Jaccard</b>	0,7206	0,2050
<b>Accuracy</b>	0,9045	0,1229

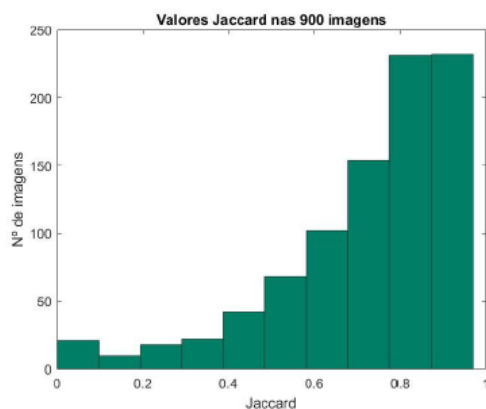


Figura 11: Valores do Jaccard nas 900 imagens

De um dataset de 900 imagens de lesão de pele, o nosso algoritmo de segmentação não foi capaz de detetar em 7 casos nenhuma lesão na imagem, o que equivale a menos de 1%. O motivo do algoritmo não ter sido bem-sucedido nestes casos é o facto de as lesões serem manchas muito ténues. É difícil, até mesmo para o olho humano detetar as bordas das lesões (Fig. 12)

e deste modo não foi possível defini-las exatamente, uma vez que as cores no interior da lesão apresentam uma pequena ou nenhuma diferença com a cor da pele circundante.

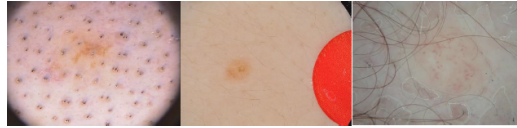


Figura 12: Imagens de lesões que não foi captada nenhuma lesão na imagem depois de implementar a segmentação (isto é, as máscaras binárias ficaram completamente a preto).

Em 114 casos (12,6%), a segmentação não foi satisfatória, uma vez que a máscara não representa a lesão completa e, portanto, algumas áreas importantes não foram abrangidas (Fig. 13).

No total existem 136 lesões que têm de ser rejeitadas devido às falhas na segmentação no conjunto das 900 imagens. O desempenho alcançado, portanto, é de cerca de 84,8% de imagens de lesão cutânea corretamente segmentadas.



Figura 13: Exemplo de uma segmentação não satisfatória. A) imagem original da lesão. B) máscara binária resultante da nossa segmentação. C) máscara binária do Ground Truth Data resultante da segmentação manual por especialistas.

## 5. Extração de Features

Nesta secção são descritos os recursos que foram utilizados para caracterizar as imagens da lesão cutânea. Um problema específico na literatura relacionada é que um número significativo de estudos não relatam os detalhes dos procedimentos de extração de características. Portanto, explicamos o fundamento de cada característica e apresentamos os aspetos algorítmicos envolvidos na computação com o máximo de detalhes possível.



### 5.1. *Features de forma*

As características globais calculadas cobrem uma ampla gama de categorias, começando pelo tamanho simples (área, perímetro, medidas polares e retângulo delimitador) e de forma (redondeza, compacidade, extensão, solidificação, excentricidade, orientação...). Outros descritores de forma são derivados do retângulo delimitador: relação de aspecto de comprimento e largura, bem como a distância entre o centro do retângulo delimitador e o centro de massa da lesão.

```
» b = regionprops(image,'Area','ConvexArea','Eccentricity','EquivDiameter',  
'Extent','FilledArea','MajorAxisLength','MinorAxisLength','Orientation',  
'Perimeter','Solidity');
```

### 5.2. *Features de cor*

Para calcular características de cor simples, as imagens originais foram transformadas do mapa de cores RGB (original) para o mapa de cores HSI, HSV e L\*a\*b. As características derivadas para cada mapa são o mínimo, o máximo, a média e a variância dos três canais dos diferentes tipos de sistemas de cor.

```

%% Aplicado a RGB image
» mean_r = sum_r/(m*n);
» min_r = min(r(r>0));
» max_r = max(r(r>0));
» var_r = var(double(r(r>0)));
(O mesmo foi executado para a componente verde e azul)

A média, o mínimo, o máximo e a variância de cada componente foi obtido
da mesma forma para os seguintes color %% Aplicado a L*a*b image
» cform = makecform('srgb2lab');
» lab = applycform(image,cform);
(O mesmo foi executado para as dimensões a e b do color space.)

%% Aplicado a HSI image
» H = H/360;
» S = 1-(3./(sum(I,3)+0.000001)).*min(I,[],3);
» I = sum(I,3)./3;
(O mesmo foi executado para a saturação e intensidade.)

%% Aplicado a HSV image
» A = rgb2hsv(image);
(O mesmo foi executado para a saturação e valor.)

```

### 5.3. Features de textura

Neste estudo foram ainda extraídas features baseadas na análise de textura usando GLCM. O GLCM é uma poderosa ferramenta para a extração de features de imagem mapeando as probabilidades de co-ocorrência de nível de cinza com base nas relações espaciais de pixels em diferentes direções angulares. A característica extraída com base no GLCM são: autocorrelação, contraste, correlação, proeminência de cluster, cluster shade, dissimilaridade, energia, entropia, homogeneidade, máxima probabilidade, variância, média de soma, variância de soma, entropia de soma, diferença de variação, diferença entropia, medida de informação de correlação, diferença inversa homogênea (INV), diferença inversa normalizada (INN) e o momento de diferença inversa normalizada [22]. Na última minha linha do código em baixo está a execução da característica da autocorrelação. As restantes características foram obtidas de maneira semelhante.

```
%% Com recurso à função GLCM_Features4.  
» GLCM2 = graycomatrix(I,'Offset',[2 0;0 2]);  
» stats = GLCM_Features4(GLCM2,0);  
» autoc = stats(1).autoc; autoc = mean(autoc);
```

Após extração de todas as features deparamo-nos com alguns valores incoerentes e não válidos (isto é, apresentavam o mesmo valor para todas as imagens) para definir a lesão e portanto foi necessário efetuar uma pré-seleção das que não apresentavam um valor possível e credível, procedendo-se à sua eliminação. Portanto no total resultaram 75 features, sendo que 11 definiam a forma da lesão, 43 definiam a cor tendo em conta os diversos colormaps e 21 definiam a textura da imagem segmentada.

## 6. Seleção de Features

O desempenho do sistema de classificação pode ser degradado usando todas as características dos dados de entrada, pois aumenta a complexidade. Otimizar os recursos é muito importante, já que alguns deles desempenham um papel mais importante no reconhecimento do que outros. Há uma série de métodos que são desenvolvidos e têm sido utilizados para selecionar características.

Numa fase inicial e para teste imediatos foi utilizando o Matlab para esta etapa, executando-se dois algoritmos distintos para verificar quais as features mais significativas, *Kruskal-Wallis* e *ReliefF*.

O algoritmo de *Kruskal-Wallis* é usado no método simples de usar e verifica se duas ou mais classes têm mediana igual e dá o valor da mesma. De acordo com o valor da mediana os recursos com informações discriminatórias são selecionados. Se o valor estiver perto de "0" significa que o recurso contém informações discriminantes; Caso contrário, ele não será selecionado. As características de DWT são processadas usando a técnica de *Kruskal-Wallis*. Os recursos que resultam desta fase são selecionados para serem usados na próxima etapa de reconhecimento.

```
» kruskal=[];  
» for i=1:length(total(1,:))  
» kruskal=kruskalwallis(total(:,i),numlables2,'off') ;  
» end
```

Utilizando este método é possível concluir que a feature com mais peso na classificação das lesões é a assimetria da mesma pois apresenta o menor valor de probabilidade de rejeição da hipótese nula, as features e as labels são dependentes. Seguidamente do nível de relevância encontra-se a saturação das cores da lesão verificada com um color map HSI.

Utilizou-se também o *ReliefF* como algoritmo de seleção para calcular as classificações e pesos dos atributos dos dados de entradas. O *ReliefF* usa K vizinhos mais próximos por classe para criar um ranking dos índices mais importantes.

» [rank,weight]=relieff(total,numlables2,1);

Com este algoritmo a feature que têm mais peso na classificação é a cor verde num colormap RGB, seguida da variância da cor vermelha e das coordenadas das cores amarelo e azul no espaço de cores  $L^*a^*b^*$ .

Seguidamente e como alternativa mais eficaz foi feita também seleção de features utilizando o software de análise de dados "Orange Canvas"(Fig. 14). Selecionaram-se features com base nos seguintes algoritmos: *PCA*, *Information Gain*, *Gain Ratio*, *Gini Decrease*, *Chi2*, *ReliefF* e *FCBF* (Tabela 2).

O procedimento estatístico PCA, utiliza uma transformação ortogonal para converter um conjunto de observações de possíveis variáveis correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas. Este método tem uma natureza completamente diferentes dos restantes algoritmos e, portanto, foi analisado separadamente.

Tabela 2: Dez features com mais peso de cada algoritmo de seleção.

Information Gain	Gain Ratio	Gini Decrease	Chi2	ReliefF	FCBF
mean_h	mean_h	mean_h	mean_h	var_r	mean_h
homop	homop	homop	homop	min_b1	Orientation
mean_g	mean_g	mean_g	mean_g	min_v	homop
max_i	max_i	max_i	energ	max_b	mean_g
dissi	dissi	dissi	senth	max_s	dissi
energ	energ	energ	dissi	var_g	max_i
senth	senth	senth	min_g	min_h	mean_b
min_r	min_r	min_r	var_h	Extent	var_i
mean_b	mean_b	min_g	max_i	corrp	min_r
var_i	var_i	var_h	min_r	max_v	senth

De forma a obter os melhores resultados, realizaram-se diversas conjugações de métodos e quantidades de features. Esta avaliação foi feita tendo em conta a matriz de confusão obtida após classificação. Devido infundável quantidade de interações não nos pareceu necessário apresentar aqui todos os valores obtidos, e portanto referimos qual a conclusão obtida após todo o estudo. Pela análise individual da implementação de cada um dos métodos (Tabela 1) verificou-se que não existiam diferenças significativas relativamente às features com maior peso na classificação das lesões e portanto a conjugação com que se obteve melhores resultados foi precisamente na média dos 6 algoritmos de seleção.

Realizaram-se também testes para averiguar a quantidade de features significativas que seria mais correto utilizar na classificação. Concluiu-se que os melhores valores de sensibilidade e especificidade eram obtidos com dez features, embora que a variação para dos resultados para valores próximos deste não causa alterações significativas.

No prosseguimento deste trabalho foram utilizados, como termo de comparação, os resultados obtidos a partir do PCA e das dez features selecionadas pelos seis restantes algoritmos. Neste último utilizaram-se então as seguintes features: a média das cores da lesão verificada com um color map HSI, a homogeneidade, a média da componente verde do colormap RGB, o máximo da componente de intensidade no colormap HSI, a dissimilaridade, a energia, a soma da entropia, o mínimo da componente vermelha do colormap RGB, a média da componente azul do colormap RGB e a variância da componente de intensidade no colormap HSI.

## 7. Classificação

No geral, extraíram-se 75 features das 764 imagens, depois de remover as imagens que não foram corretamente segmentadas (136). Destas 764 imagens, 618 representam melanomas benignos e 146 melanomas malignos.

Os resultados foram classificados por duas maneiras diferentes. Recorrendo primeiro a classificadores convencionais com recurso ao software de análise de dados *Orange Canvas* e, de seguida, a classificadores usando redes e Machine Learning recorrendo ao Matlab para o efeito.

### 7.1. Classificadores convencionais

Os **classificadores convencionais** que utilizámos foram o *Classification Tree*, *Logistic Regression*, *Naive Bayes*, *Nearest Neighbors*, *Random Forest Classification* e *SVM*, utilizando o *cross validation* para testar os classificadores.

Para utilizar o programa basta clicar no ícone *File*, fazer download do ficheiro *features\_764.csv* e depois na **feature 76** alterar de *feature* para *target*.

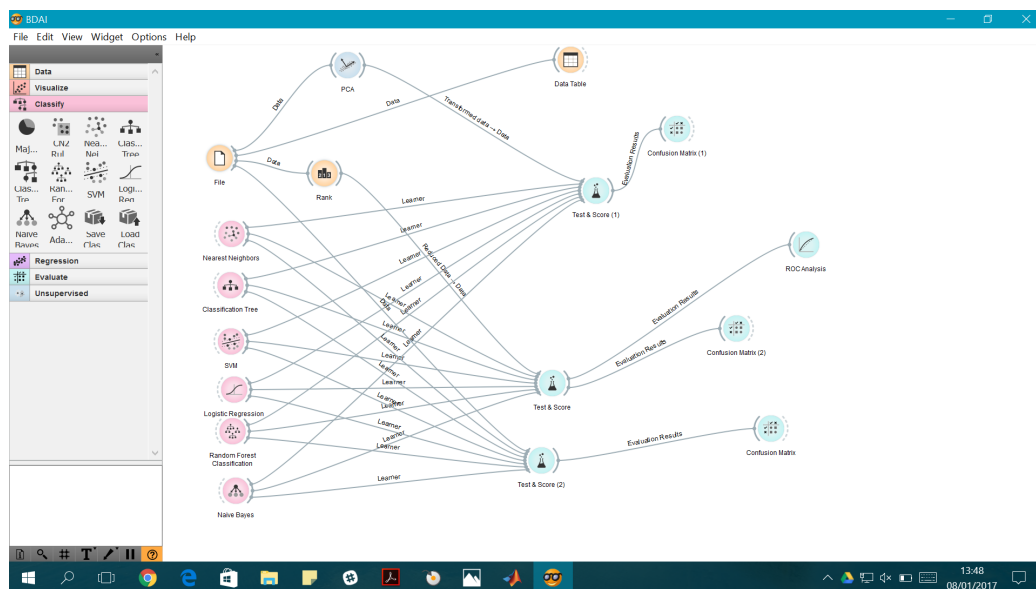


Figura 14: Representação dos classificadores no programa utilizado, Orange Canvas.

## 7.2. Outros Classificadores

Para além dos classificadores convencionais, decidimos ir mais além e classificar as nossas imagens através do treino e teste de uma rede de classificação de categorias de imagens recorrendo à ***Statistics and Machine Learning Toolbox do Matlab***. Este tipo de classificação já foi executado no Matlab.

Primeiro foi se buscar as imagens que foram previamente divididas em duas pastas, designadas por "malignos" e "benignos". De seguida, criamos um *datastore* das imagens especificadas pela sua localização e com os labels referentes ao nome da pasta em que estavam inseridos. Um *datastore* é uma repositório de coleção de dados normalmente utilizado quando os dados são demasiado grandes para a memória suportar. Ele não faz download das imagens, executa sim as funções na localização dos ficheiros pretendidos.

A função *countEachLabel* foi utilizada só para verificar se as imagens estavam bem identificadas por os labels. Esta função retorna o número de imagens de cada label.

```
»myFolder= fullfile(pwd);
»filePattern = fullfile(myFolder, '*.jpg');
»jpegFiles = dir(filePattern);
»categorias = {'malignos', 'benignos'};
»imds = imageDatastore(fullfile(myFolder, categorias), 'LabelSource', 'foldernames');
»tbl = countEachLabel(imds)
```

Como o data set acima referido contém um número desigual de imagens por categoria, primeiro ajustou-se esse número (igualamos número de imagens benignas (727) às malignas (173)) através da função *splitEachLabel*, para depois dividir o data set no data set do treino e no de validação.

Fizeram-se várias tentativas para descobrir qual a melhor divisão a fazer do data set. Os melhores resultados obtidos foram quando se dividiu-se o data set em 50% para treino e os outros 50% para teste e validação.

O *randomize* é utilizado para dividir os dados de forma aleatória para evitar que influenciar os resultados obtidos.

```

»minSetCount = min(tbl,2);
»imds = splitEachLabel(imds, minSetCount, 'randomize');
»countEachLabel(imds)
»[trainingSet, validationSet] = splitEachLabel(imds, 0.5, 'randomize');

```

Nesta classificação calcularam-se outras features, recorrendo à função *bag-features*. Esta função extrai *SURF features* de todas as imagens das duas categorias e constrói uma "vocabulário visual" reduzindo o número de *features* através da quantização do espaço de *features* usando a agregação de *K-means clustering*.

```

»bag = bagOfFeatures(trainingSet);

```

Para finalizar, treinou-se a rede com o data set do treino e as features obtidas em cima, através da função *trainImageCategoryClassifier* e avaliou-se os resultados obtidos pela matriz de confusão para o *trainingSet* e para o *validationSet* que não foi utilizado no treino.

```

»categoryClassifier = trainImageCategoryClassifier(trainingSet, bag);
»confMatrix = evaluate(categoryClassifier, trainingSet);
»confMatrix = evaluate(categoryClassifier, validationSet);

```

Depois da rede treinada e guardada, podemos utiliza-la para classificar qualquer imagem que queiramos utilizando a função *predict*.

```

»[filename,pathname]=uigetfile('*.jpg');
»fullFileName = fullfile(pathname, filename);
»img = imread(fullFileName);
»img = imread(fullfile(myFolder, 'malignos', 'ISIC_ 0000140.jpg'));
»[labelIdx, scores] = predict(categoryClassifier, img);
»result =string (categoryClassifier.Labels(labelIdx));
»display(strcat('A classificação da imagem é: ',result));

```

Também classificamos as nossas imagens, recorrendo à **Neural Network Toolbox** do Matlab (código encontra-se no ficheiro designado por "NN.m"). O ficheiro "total\_features.mat" contém as features das 764 imagens, e o fi-



cheiro "labels" as categorias das mesmas 764 imagens. Esta rede classifica as imagens através do treino e teste de uma rede neuronal designada por fitnet (função fitting neural network) usando a função *Levenberg-Marquardt* para treinar a rede e uma camada escondida de 22 neurónios.

```
»trainFcn = 'trainlm';  
»hiddenLayerSize = 22;  
»net = fitnet(hiddenLayerSize,trainFcn);
```

Dividiu-se o data set em 70% para treino, 15% para validação e 15% para test. Treinamos e testamos a rede usando as funções *train* e *perform*, respetivamente.

```
»net.divideParam.trainRatio = 70/100;  
»net.divideParam.valRatio = 15/100;  
»net.divideParam.testRatio = 15/100;  
»[net,tr] = train(net,x,t);  
»y = net(x);  
»e = gsubtract(t,y);  
»performance = perform(net,t,y);
```

Para terminar, normalizaram-se os dados obtidos pelo output para posteriormente ser possível calcular as métricas.

```
»for i=1:length(e)  
»if e(i)<0.51  
»normalizado(i,1)=0;  
»else  
»normalizado(i,1)=1;  
»end  
» end
```

No final, para avaliar os vários classificadores, para posterior análise, calcularam-se os valores de especificidade e sensibilidade.

### 7.3. Resultados

Todos os algoritmos convencionais referidos foram testados para as features selecionadas pelo método de PCA e as features obtidas a partir da média dos seis métodos. Para termos comparativos classificou-se ainda todas as features obtidas, sem qualquer seleção, desta forma podemos verificar as alterações que a otimização de features provoca.

Tabela 3: Resultados de sensibilidade e especificidade para os diferentes classificadores convencionais e de acordo com as features utilizadas.

	<b>Total de Features</b>		<b>Média Rank</b>		<b>PCA</b>	
	Sens.	Esp.	Sens.	Esp.	Sens.	Esp.
<b>K-Nearest Neighbors</b>	0,947	0,089	0,935	0,103	0,924	0,103
<b>Decision Tree</b>	0,854	0,329	0,830	0,253	0,867	0,178
<b>SVM</b>	0,866	0,281	0,626	0,425	0,435	0,486
<b>Logistic Regression</b>	0,977	0,089	0,992	0,034	0,994	0,014
<b>Random Classification</b>	0,945	0,151	0,951	0,068	0,861	0,158
<b>Naive Bayes</b>	0,571	0,712	0,586	0,685	1	0

Para avaliação dos diferentes classificadores calculou-se o valor da sensibilidade e especificidade. A sensibilidade reflete o quanto o método é eficaz em identificar corretamente, dentro todos os indivíduos avaliados, aqueles que realmente apresentam melanomas benignos. Já a especificidade de um método reflete o quanto ele é eficaz em identificar corretamente os indivíduos que apresentam melanomas malignos.

É bom ter valores altos de ambos os parâmetros mencionados em cima, mas o parâmetro mais relevante neste caso é a especificidade pois é essencial detetar corretamente todos os casos de melanoma malignos.

Seguidamente sintetizamos os resultados obtidos na *Tabela 3* e contextualizamos no âmbito do projeto:

1. *Nearest Neighbors* - Este classificador é o pior a classificar os melanomas malignos, pois deteta todos as imagens como sendo benignas.
2. *Classification Tree* - Neste método, o melhor resultado obteve-se quando se consideraram todas as features, no entanto, mesmo assim os resultados obtidos para a especificidade não foram bons o suficiente.
3. *SVM* - Obteve-se uma melhor sensibilidade na totalidade das features, mas de um modo geral o melhor resultado observou-se na média dos Ranks.

4. *Logistic Regression* - Este algoritmo foi o classificador que obteve o valor mais alto para a Sensibilidade, no entanto obteve valores muito baixos para a especificidade, o que torna, este classificador péssimo em semelhança ao primeiro (Nearest Neighbors).
5. *Random Forest Classification* - Este classificador obteve resultados péssimos no geral, semelhantes ao primeiro e quarto classificadores.
6. *Naive Bayes* - Este classificador, que assume que os atributos são condicionalmente independentes, foi de todos o que obteve melhores resultados, pois obteve o maior valor de especificidade entre todos e obteve um valor razoável de sensibilidade, exceto quando a seleção de features foi efectuada pelo PCA

Os resultados obtidos pelos os outros 2 algoritmos através das Toolboxes do Matlab estão representados em baixo.

Tabela 4: Resultados de sensibilidade e especificidade para os 2 classificadores recorrendo ao Matlab.

	Sens.	Esp.
<b>Image Category Classifier</b>	0,74	0,75
<b>Neural Network</b>	0,91	1

Estes 2 algoritmos apresentam resultados muito melhores comparativamente aos outros tipos de classificadores, sendo que a rede neuronal apresenta de longe os melhores resultados.

## 8. Discussão

O Data set fornecido para o concurso "*ISBI 2016: Skin Lesion Analysis Towards Melanoma Detection*" que pertencem ao arquivo do ISIC (*International Skin Imaging Collaboration*), composto por 900 imagens dermatoscópicas divididas em 2 classes, melanomas benignos e malignos.

### 8.1. Segmentação

No geral, podemos concluir que realizámos uma segmentação aceitável, pois obtivemos mais de 84% de imagens corretamente segmentadas comparando com o Ground Truth fornecido pelo o concurso. A performance da

segmentação é conseguida através da fusão de vários algoritmos de segmentação, como por exemplo o *thresholding*. O valor de imagens corretamente segmentadas não é maior devido à grande diversidade de objetos e pêlos no nosso *data set* o que torna a pré-processamento e a segmentação uma tarefa muito árdua.

### 8.2. Extração e Seleção de Features

Relativamente à extração de features, o principal foco das features extraídas a partir das imagens já segmentadas foram os parâmetros de formas e cores.

Através da seleção de features efetuada, o número dos parâmetros mais relevantes para uma boa classificação foi reduzido de 75 para 10 parâmetros. Observou-se que a seleção de features não melhorou assim tanto os resultados, como era esperado acontecer. Os melhores resultados foram obtidos usando as features extraídas pela função *bagOfFeatures*, sendo que é uma parte do trabalho que merece ser mais aprofundado para melhorias futuras.

### 8.3. Classificação

Os resultados obtidos para a classificação utilizando classificadores convencionais não são assim tão bons, devido provavelmente ao facto do *data set* utilizado não ser muito extenso, tal como a fração de imagens de melanomas malignos ser muito reduzida comparativamente ao total de imagens fornecidas (19%), o que faz com que os classificadores fiquem bem treinados para classificar os melanomas benignos e maus para classificar os malignos.

Os 2 outros classificadores, em que o primeiro utilizou um conjunto de features próprios do método e o outro utilizou a totalidade das 75 features extraídas pelos métodos que realizamos, obtiveram resultados muito melhores, obtendo-se os melhores resultados nas redes neuronais. No entanto, este só utiliza 15% do *data set* para testar o classificador, sendo que não foi possível tirar conclusões muito fidedignas. De uma maneira geral e analisando todos os resultados obtidos concluímos que através do *Image Category Classifier* obtêm-se uma classificação mais fidedigna e real, obtendo uma *accuracy* de 75%.

Para finalizar, o nosso classificador é capaz de detetar corretamente uma ampla quantidade de lesões de pele em malignos e benignos. No entanto, há aspetos que devem ser tidos em conta. O nosso *data set* é relativamente pequeno, para tirar conclusões mais fidedignas deveria-se aumentar o mesmo. Podemos verificar que o nosso melhor classificador não utilizou as 75 *features*

extraídas por nós, sendo que pode ser uma razão para os classificadores convencionais não darem tão bons resultados. Para obter melhores resultados na classificação, novos parâmetros devem ser incluídos nas *features*. Parâmetros que permitam a descrição entre a transição da área entre a lesão e a pele não estão incluídos nas *features* que obtivemos e testamos, e muito provavelmente é um bom ponto de partida para melhorar os resultados finais.

## 9. Conclusão

As imagens que compõem a nossa base de dados são bastantes difusas e heterogêneas visualmente e, para a vista normal sem experiência clínica, muito difíceis de distinguir os dois tipos de melanomas. Ao criar-se um método automático, que consiga classificar uma imagem dermatológica como maligno ou benigno, são várias as vantagens que daí advém nomeadamente não estar dependente de uma avaliação de dermatologistas cujo número de avaliações que consegue dar são limitadas ao seu tempo disponível. Considerando que neste trabalho a distinção é entre 2 tipos diferentes de melanoma, o valor de desempenho que conseguimos, 75% através do uso de uma rede baseada em *Machine Learning* é um valor bastante aceitável e encorajador para prosseguir e aperfeiçoar esta técnica em abordagens futuras.

## Referências

- [1] H. W. Rogers, M. A. Weinstock, S. R. Feldman, B. M. Coldiron, Incidence estimate of nonmelanoma skin cancer (keratinocyte carcinomas) in the us population, 2012, JAMA dermatology 151 (2015) 1081–1086.
- [2] M. S. Brady, S. A. Oliveria, P. J. Christos, M. Berwick, D. G. Coit, J. Katz, A. C. Halpern, Patterns of detection in patients with cutaneous melanoma, Cancer 89 (2000) 342–347.
- [3] H. Kittler, H. Pehamberger, K. Wolff, M. Binder, Diagnostic accuracy of dermoscopy, The lancet oncology 3 (2002) 159–165.
- [4] M. Binder, M. Schwarz, A. Winkler, A. Steiner, A. Kaider, K. Wolff, H. Pehamberger, Epiluminescence microscopy: a useful tool for the diagnosis of pigmented skin lesions for formally trained dermatologists, Archives of Dermatology 131 (1995) 286–291.

- [5] M. Vestergaard, P. Macaskill, P. Holt, S. Menzies, Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: a meta-analysis of studies performed in a clinical setting, *British Journal of Dermatology* 159 (2008) 669–676.
- [6] Metaoptima, Molescope<sup>TM</sup>, ????
- [7] T. Mendonca, A. R. Marcal, A. Vieira, J. C. Nascimento, M. Silveira, J. S. Marques, J. Rozeira, Comparison of segmentation methods for automatic diagnosis of dermoscopy images, in: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, pp. 6572–6575.
- [8] A. Green, N. Martin, J. Pfitzner, M. O’Rourke, N. Knight, Computer image analysis in the diagnosis of melanoma, *Journal of the American Academy of Dermatology* 31 (1994) 958–964.
- [9] G. A. Hance, S. E. Umbaugh, R. H. Moss, W. V. Stoecker, Unsupervised color image segmentation: with application to skin tumor borders, *IEEE Engineering in Medicine and Biology Magazine* 15 (1996) 104–111.
- [10] S. E. Umbaugh, R. H. Moss, W. V. Stoecker, G. Hance, Automatic color segmentation algorithms-with application to skin tumor feature identification, *IEEE Engineering in Medicine and Biology Magazine* 12 (1993) 75–82.
- [11] H. Ganster, P. Pinz, R. Rohrer, E. Wildling, M. Binder, H. Kittler, Automated melanoma recognition, *IEEE transactions on medical imaging* 20 (2001) 233–239.
- [12] B. Erkol, R. H. Moss, R. Joe Stanley, W. V. Stoecker, E. Hvatum, Automatic lesion boundary detection in dermoscopy images using gradient vector flow snakes, *Skin Research and Technology* 11 (2005) 17–26.
- [13] M. Silveira, J. C. Nascimento, J. S. Marques, A. R. Marçal, T. Mendonça, S. Yamauchi, J. Maeda, J. Rozeira, Comparison of segmentation methods for melanoma diagnosis in dermoscopy images, *IEEE Journal of Selected Topics in Signal Processing* 3 (2009) 35–45.

- [14] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, R. H. Moss, A methodological approach to the classification of dermoscopy images, *Computerized Medical Imaging and Graphics* 31 (2007) 362–373.
- [15] C. Barata, M. Ruela, M. Francisco, T. Mendonça, J. S. Marques, Two systems for the detection of melanomas in dermoscopy images using texture and color features, *IEEE Systems Journal* 8 (2014) 965–979.
- [16] R. Archibald, G. Fann, Feature selection and classification of hyperspectral images with support vector machines, *IEEE Geoscience and remote sensing letters* 4 (2007) 674–677.
- [17] N. Cascinelli, M. Ferrario, T. Tonelli, E. Leo, A possible new tool for clinical diagnosis of melanoma: the computer, *Journal of the American Academy of Dermatology* 16 (1987) 361–367.
- [18] M. E. Celebi, H. Iyatomi, W. V. Stoecker, R. H. Moss, H. S. Rabinovitz, G. Argenziano, H. P. Soyer, Automatic detection of blue-white veil and related structures in dermoscopy images, *Computerized Medical Imaging and Graphics* 32 (2008) 670–677.
- [19] A. P. James, B. V. Dasarathy, Medical image fusion: A survey of the state of the art, *Information Fusion* 19 (2014) 4–19.
- [20] P. Rubegni, G. Cevenini, M. Burrioni, R. Perotti, G. Dell’Eva, P. Sbano, C. Miracco, P. Luzzi, P. Tosi, P. Barbini, et al., Automated diagnosis of pigmented skin lesions, *International Journal of Cancer* 101 (2002) 576–580.
- [21] P. Rubegni, G. Cevenini, M. Burrioni, R. Perotti, G. Dell’Eva, P. Sbano, C. Miracco, P. Luzzi, P. Tosi, P. Barbini, et al., Automated diagnosis of pigmented skin lesions, *International Journal of Cancer* 101 (2002) 576–580.
- [22] M. A. Sheha, M. S. Mabrouk, A. Sharawy, Automatic detection of melanoma skin cancer using texture analysis, *International Journal of Computer Applications* 42 (2012) 22–26.