



UNIVERSIDADE DE ÉVORA

Mestrado em Engenharia Informática

Processamento de Imagem Digital para o Controlo de
Qualidade do Queijo Regional de Évora

Gaspar Manuel Rocha Brogueira

Orientador: *Doutor Luís Miguel de Mendonça Rato*



177943

Julho de 2010

Resumo

A utilização generalizada do computador para a automatização das mais diversas tarefas, tem conduzido ao desenvolvimento de aplicações que possibilitam a realização de actividades que até então poderiam não só ser demoradas, como estar sujeitas a erros inerentes à actividade humana.

A investigação desenvolvida no âmbito desta tese, tem como objectivo o desenvolvimento de um software e algoritmos que permitam a avaliação e classificação de queijos produzidos na região de Évora, através do processamento de imagens digitais.

No decurso desta investigação, foram desenvolvidos algoritmos e metodologias que permitem a identificação dos olhos e dimensões do queijo, a presença de textura na parte exterior do queijo, assim como características relativas à cor do mesmo, permitindo que com base nestes parâmetros possa ser efectuada uma classificação e avaliação do queijo.

A aplicação de software, resultou num produto de simples utilização. As fotografias devem respeitar algumas regras simples, sobre as quais se efectuará o processamento e classificação do queijo.

Palavras Chave: Processamento de Imagem, Processamento de Sinal, Segmentação, Textura, Classificação, Aprendizagem Automática, Queijo de Évora.

Abstract

Digital Image Processing Applied to Regional Évora Cheese Quality Control

The widespread use of computers for the automation of repetitive tasks, has resulted in developing applications that allow a range of activities, that until now could not only be time consuming and also subject to errors inherent to human activity, to be performed without or with little human intervention.

The research carried out within this thesis, aims to develop a software application and algorithms that enable the assessment and classification of cheeses produced in the region of Évora, by digital images processing.

Throughout this research, algorithms and methodologies have been developed that allow the identification of the cheese eyes, the dimensions of the cheese, the presence of texture on the outside of cheese, as well as an analysis of the color, so that, based on these parameters, a classification and evaluation of the cheese can be conducted.

The developed software application, is product simple to use, requiring no special computer knowledge. Requires only the acquisition of the photographs following a simple set of rules, based on which it will do the processing and classification of cheese.

Keywords: Image Processing, Signal Processing, Segmentation, Texture, Classification, Machine Learning, Cheese of Évora.

Agradecimentos

Gostaria de expressar o meu profundo reconhecimento ao meu orientador, o Prof. Dr. Luís Rato por todo o empenho na aquisição das indispensáveis fotografias dos queijos, pelo enorme apoio concedido e pelas sempre estimulantes sugestões e apreciações críticas.

Um reconhecimento especial à Eng. Maria da Graça Janeiro Machado, responsável pelo Laboratório de Tecnologia e Qualidade dos Produtos Naturais, na Universidade de Évora, pelas produtivas conversas sobre a classificação e avaliação do queijo regional de Évora.

Palavras de apreço são também devidas ao Prof. Dr. Miguel Barão, pelos seus esclarecimentos relativos às técnicas de detecção de texturas em imagens digitais.

Ao Prof. José Saias pela inteira disponibilidade em sugerir, discutir e apreciar alguns aspectos da implementação do software desenvolvido ao longo desta tese, expresse o meu agradecimento.

À Sara Isabel Santos Gonçalves por todo o tempo dispendido sempre que necessário e pela inexcedível ajuda. Obrigado pelo teu apoio e compreensão!

Para a minha família, um muito especial agradecimento por todo apoio e estímulo que me deram no decorrer de todo o trabalho.

Obrigado Pai, Mãe, Salomé e Avós!

Termino agradecendo a dois amigos. Obrigado Luís Borrego e Bruno Figueiredo!

Prefácio

Este documento contém uma dissertação intitulada Processamento de Imagem Digital para o controlo de qualidade do Queijo Regional de Évora.

O autor do trabalho é licenciado em Engenharia Informática, pela Universidade de Évora. A presente dissertação foi entregue em Julho de 2010.

Conteúdo

Resumo	i
Abstract	iii
Agradecimentos	v
Prefácio	vii
Lista de Figuras	xiii
Lista de Tabelas	xvii
Nomenclatura	xix
Abreviaturas	xxi
1 Introdução	1
1.1 Motivação	1
1.2 Contributos	3
1.3 Estrutura da tese	3
2 Estado da Arte	5
3 Sinais, Sistemas e Filtros Digitais	9
3.1 Introdução	9
3.2 Sinais digitais	9
3.3 Sistemas digitais	11
3.4 Convolução	12
3.5 Filtros digitais	13
4 Aquisição e Pré-processamento das Imagens	15
4.1 Introdução	15

4.2	Definição do ambiente	15
4.3	Detecção da moldura	16
4.4	Eliminação de ruído	19
4.5	Rotação da imagem	20
4.6	Reconhecimento do queijo	22
4.6.1	Conversão de RGB para HSV	22
4.6.2	Separação do queijo da moldura	23
4.6.3	Cálculo das dimensões do queijo	23
5	Segmentação	25
5.1	Introdução	25
5.2	Apresentação do algoritmo desenvolvido	25
5.3	Considerações sobre o algoritmo	31
5.3.1	Número de iterações da filtragem não linear	31
5.3.2	Comportamento oscilante	32
5.3.3	Valores da resposta impulsiva nos limites da imagem	33
5.4	Resumo do processo de segmentação	33
5.4.1	Representação em diagrama de blocos	34
5.5	Resultados	34
5.6	Comparação com outros algoritmos de segmentação	36
5.6.1	<i>Otsu Threshold</i>	36
5.6.2	<i>Maximum Entropy Threshold</i>	37
5.6.3	<i>K-Means Clustering</i>	38
6	Detecção de Texturas	41
6.1	Introdução	41
6.2	Textura	41
6.3	Transformada de Hough	42
6.3.1	Transformada de Hough para detecção de rectas	42
6.3.2	Transformada de Hough para outras figuras geométricas	45
6.3.3	Algoritmo da transformada de Hough	46
6.4	Metodologia adoptada para detecção de textura	47
6.4.1	Função de avaliação	48
6.4.2	Escolha do limiar	49
6.4.3	Algoritmo implementado	49
6.5	Resultados	50
6.6	Abordagem inicial	50
6.6.1	Cálculo do centro de massa	51
6.6.2	Circunferência centrada no centro de massa	52

7	Recolha de Atributos	55
7.1	Introdução	55
7.2	Atributos relativos aos olhos	55
7.2.1	Contagem na diagonal	59
7.3	Atributos relativos à cor	61
8	Aprendizagem e Classificação	63
8.1	Introdução	63
8.2	Aprendizagem	63
8.3	Avaliação do especialista	65
8.4	Modelo de aprendizagem	65
8.5	Modelo de classificação	67
8.5.1	Algoritmo de <i>Stacking</i>	67
8.5.2	<i>Nível-0</i>	68
8.5.2.1	Classificação em função da cor - Algoritmo <i>Sequential Minimal Optimization</i>	68
8.5.2.2	Classificação em função dos olhos - Algoritmo J48	69
8.5.2.3	Classificação em função da homogeneidade da cor - Algoritmo <i>Best-First Decision Tree</i>	71
8.5.2.4	Classificação em função da cor e dos olhos - Algoritmo <i>Alternating Decision Tree</i>	74
8.5.3	<i>Nível-1</i>	76
8.5.3.1	Algoritmo <i>RandomTree</i> - Esquema de <i>Stacking</i>	76
8.5.3.2	Análise dos resultados	78
8.5.3.3	Árvore de classificação final	79
8.5.3.4	Resumo das classificações do <i>nível-1</i>	79
8.6	Abordagem alternativa	81
9	Aplicação de Software	83
9.1	Introdução	83
9.2	Interface de utilizador	83
9.2.1	Modos de utilização	84
9.2.2	Visualização dos resultados	85
9.2.3	Análise de desempenho	85
9.2.4	Ficheiros criados	87
9.2.5	Documentação	88
9.3	Software de desenvolvimento	88
10	Conclusão	91
10.1	Perspectivas futuras	91

Bibliografia	93
A Apêndice ao capítulo: Segmentação	99
A.1 Resultados do algoritmo de segmentação	99
A.2 Valores da segmentação para cada filtro	106
A.3 Limiares determinados pelo método de Otsu	108
B Apêndice ao capítulo: Detecção de Texturas	109
B.1 Resultados do algoritmo de detecção de texturas	109
B.2 Valores da detecção de textura	113
C Apêndice ao capítulo: Recolha de Atributos	115
C.1 Lista de atributos de cada imagem	115
C.2 Características dos olhos pelo algoritmo <i>Analyze Particles</i>	120
D Apêndice ao Capítulo: Aprendizagem e Classificação	123
D.1 Resumo das classificações do <i>nível-0</i>	123
D.2 Resumo das classificações do <i>nível-1</i>	123

Lista de Figuras

3.1	Exemplo de sinais.	10
3.2	Sinal impulso unitário de Kronecker.	10
3.3	Representação do impulso de Dirac a duas dimensões $\delta(u, v)$	11
4.1	Exemplos de imagens aceites pela aplicação.	16
4.2	Exemplos de imagens incorrectas.	17
4.3	Detecção da moldura com os operadores de Prewitt.	19
4.4	Esquema da aplicação do filtro de mediana.	19
4.5	Aplicação do filtro de mediana.	20
4.6	Determinação dos pontos para o cálculo do ângulo de rotação.	21
4.7	Representação dos espaços de cor RGB e HSV.	23
4.8	Distinção entre a região de cores do fundo da moldura e do queijo.	23
5.1	Imagem visualizada em cada uma das três componentes de cor RGB.	26
5.2	Variação de cada uma das componentes de cor para uma linha da imagem 5.1a.	26
5.3	Diferença da variação de cor em sinais com e sem a presença de olhos.	26
5.4	Resultado da aplicação do filtro $H_{1/6}$ a uma linha da imagem 5.1a.	28
5.5	Resultado da aplicação do filtro linear.	29
5.6	Aplicação do filtro $H_{1/6}$ na forma não linear.	31
5.7	Relação entre o número de iterações do filtro não linear e o acréscimo de área reconhecida como olhos do queijo.	32
5.8	Imagens em que as iterações dos filtros não lineares provocam um comportamento oscilante.	32
5.9	Gráfico ilustrativo da oscilação no resultado da segmentação das fotografias 5.8a e 5.8b.	33
5.10	Diagrama de blocos que resume o algoritmo de segmentação.	34
5.11	Segmentação da imagem 2521e.jpg.	34
5.12	Segmentação da imagem 8895b.jpg.	35
5.13	Segmentação da imagem 7843b.jpg.	35
5.14	Segmentação da imagem 5946a.jpg.	35
5.15	Aplicação do algoritmo <i>Otsu Threshold</i>	36
5.16	Comparação da segmentação desenvolvida com o método de Otsu.	37

5.17	5.17b, 5.17d, 5.17f segmentação com o algoritmo <i>Maximum Entropy Threshold</i> , das imagens 5.17a, 5.17c, 5.17e, respectivamente.	38
5.18	5.18b, 5.18d, 5.18f segmentação com o algoritmo <i>K-Means Clustering</i> , das imagens 5.18a, 5.18c, 5.18e, respectivamente.	39
6.1	Definição gráfica de pontos colineares.	43
6.2	Conjunto de rectas que passam num ponto.	43
6.3	Relação entre o espaço da imagem e o espaço de parâmetros.	44
6.4	Representação de linhas em 2D.	44
6.5	Exemplos de queijos que apresentam textura na sua crosta.	47
6.6	Aplicação dos operadores de Sobel.	48
6.7	Aplicação da transformada de Hough.	48
6.8	Gráfico da função <i>coseno</i>	49
6.9	Gráfico da função de avaliação.	49
6.10	Resultado da aplicação do algoritmo para o cálculo do centro de massa.	51
6.11	Sinal RGB dos valores amostrados para circunferências com raios diferentes, centradas no centro de massa.	53
7.1	Aplicação do <i>plugin Analyze Particles</i> , do ImageJ.	56
7.2	Análise da distribuição do tamanho dos olhos.	57
7.3	Exemplificação das direcções de contagem referentes às características dos olhos.	58
7.4	Gráficos de dispersão que demonstram a relação entre olhos pequenos e olhos grandes.	59
7.5	Definição das regiões onde serão recolhidos parâmetros de cor.	61
8.1	Definição e atribuição de classe a cada queijo do <i>corpus</i> e instâncias de transição.	66
8.2	Matriz de confusão do algoritmo <i>Sequential Minimal Optimization</i>	70
8.3	Matriz de confusão do algoritmo J48.	71
8.4	Matriz de confusão do algoritmo <i>Best-First Decision Tree</i>	74
8.5	Matriz de confusão do algoritmo <i>Alternating Decision Tree</i>	76
8.6	Matriz de confusão do algoritmo RandomTree.	77
8.7	Matriz de confusão 2×2	78
8.8	Valores das métricas de avaliação do modelo de <i>Stacking</i>	78
8.9	Esquema da árvore de decisão gerada pelo algoritmo de <i>Stacking</i>	80
8.10	Matriz de confusão do algoritmo RandomForest.	81
8.11	Valores das métricas de avaliação do algoritmo RandomForest.	82
9.1	Aspecto visual da interface.	84
9.2	Formas de avaliar as fotografias.	85
9.3	Tabela com os dados da detecção dos olhos do queijo, apresentada na interface.	85

9.4	Tabela com os dados da avaliação do queijo, apresentada na interface. . . .	86
9.5	Análise do desempenho do software.	86
9.6	Exemplo da estrutura do ficheiro XML gerado após o processamento e avaliação de uma fotografia.	87
9.7	Ficheiros criados ao longo da avaliação de uma imagem.	88
9.8	Documentação gerada pela ferramenta Javadoc.	89
9.9	Esquema que resume o software utilizado para o desenvolvimento da aplicação.	89
A.1	Segmentação da imagem 6954a.jpg (1).	99
A.2	Segmentação da imagem 2521d.jpg (2).	99
A.3	Segmentação da imagem 3124a.jpg (3).	99
A.4	Segmentação da imagem 6797a.jpg (4).	100
A.5	Segmentação da imagem 2920a.jpg (5).	100
A.6	Segmentação da imagem 2876a.jpg (6).	100
A.7	Segmentação da imagem 9159a.jpg (7).	100
A.8	Segmentação da imagem 4936a.jpg (8).	100
A.9	Segmentação da imagem 3124b.jpg (9).	101
A.10	Segmentação da imagem 4190a.jpg (10).	101
A.11	Segmentação da imagem 6797c.jpg (11).	101
A.12	Segmentação da imagem 5946a.jpg (12).	101
A.13	Segmentação da imagem 9874a.jpg (13).	101
A.14	Segmentação da imagem 8895b.jpg (14).	102
A.15	Segmentação da imagem 8373a.jpg (15).	102
A.16	Segmentação da imagem 7843b.jpg (16).	102
A.17	Segmentação da imagem 5437a.jpg (17).	102
A.18	Segmentação da imagem 4856a.jpg (18).	102
A.19	Segmentação da imagem 8352b.jpg (19).	103
A.20	Segmentação da imagem 7091a.jpg (20).	103
A.21	Segmentação da imagem 2521c.jpg (21).	103
A.22	Segmentação da imagem 7843a.jpg (22).	103
A.23	Segmentação da imagem 4856b.jpg (23).	104
A.24	Segmentação da imagem 2103a.jpg (24).	104
A.25	Segmentação da imagem 8895a.jpg (25).	104
A.26	Segmentação da imagem 6797b.jpg (26).	104
A.27	Segmentação da imagem 7091b.jpg (27).	104
A.28	Segmentação da imagem 2920b.jpg (28).	105
A.29	Segmentação da imagem 9159c.jpg (29).	105
A.30	Segmentação da imagem 4936b.jpg (30).	105
A.31	Segmentação da imagem 9159b.jpg (31).	105
A.32	Segmentação da imagem 8352a.jpg (32).	105
A.33	Segmentação da imagem 7540a.jpg (33).	106

A.34 Segmentação da imagem 2521e.jpg (34).	106
B.1 Resultado da aplicação da transformada de Hough, imagem 1.	109
B.2 Resultado da aplicação da transformada de Hough, imagem 2.	109
B.3 Resultado da aplicação da transformada de Hough, imagem 3.	110
B.4 Resultado da aplicação da transformada de Hough, imagem 4.	110
B.5 Resultado da aplicação da transformada de Hough, imagem 5.	110
B.6 Resultado da aplicação da transformada de Hough, imagem 6.	110
B.7 Resultado da aplicação da transformada de Hough, imagem 7.	110
B.8 Resultado da aplicação da transformada de Hough, imagem 8.	111
B.9 Resultado da aplicação da transformada de Hough, imagem 9.	111
B.10 Resultado da aplicação da transformada de Hough, imagem 10.	111
B.11 Resultado da aplicação da transformada de Hough, imagem 11.	111
B.12 Resultado da aplicação da transformada de Hough, imagem 12.	111
B.13 Resultado da aplicação da transformada de Hough, imagem 13.	112
B.14 Resultado da aplicação da transformada de Hough, imagem 14.	112
B.15 Resultado da aplicação da transformada de Hough, imagem 15.	112

Lista de Tabelas

6.1	Relação entre o espaço da imagem e o espaço de parâmetros de Hough. . . .	44
7.1	Valores da contagem dos pixels referentes aos olhos, nas três direcções, para os exemplos da figura 7.3.	58
A.1	Resultado do algoritmo de segmentação nas diversas iterações.	107
A.2	Valores dos limiares parciais e limiar total pelo método de Otsu.	108
B.1	Resultados da aplicação da transformada de Hough a diversas imagens. . .	113
C.1	Resultado da extracção das características dos olhos pelo <i>plugin Analyze Particles</i>	121
D.1	Resumo do comportamento de cada classificador do <i>nível-0</i>	124
D.2	Resultado das classificações do <i>nível-1</i>	125

Nomenclatura

- β Constante arbitrária
- \otimes Operador de convolução
- \ominus Direcção do vector gradiente
- H Matriz $n \times n$, correspondente à máscara de um filtro
- R Região de suporte da máscara de um filtro
- K Número de pontos do sinal de entrada usados no filtro de média móvel
- $E(X)$ Entropia da variável X
- $h(n)$ Resposta impulsiva discreta no instante n
- $y(n)$ Sinal de saída discreto no instante n
- $x(n)$ Sinal de entrada discreto no instante n
- (\tilde{x}, \tilde{y}) Coordenadas do centro de massa
- $I(i, j)$ Imagem I com i linhas e j colunas
- $I'(i, j)$ Imagem resultante da aplicação de um filtro linear
- $\delta(n)$ Sinal impulso unitário, ou impulso unitário de Dirac
- $\delta(u, v)$ Sinal impulso unitário a duas dimensões
- $F_T(I(i, j), \tau)$ Função de limiarização da imagem I com o limiar τ
- $\nabla I(i, j)$ Vector gradiente da imagem $I(i, j)$
- $H_{1/6}$ Filtro de média móvel com $K = 1/6$ pontos da largura da imagem
- $a(n)$ Resposta impulsiva do filtro $H_{1/6}$

- $y_a(n)$ Sinal resultante da aplicação do filtro $H_{1/6}$
- $H_{1/60}$ Filtro de média móvel com $K = 1/60$ pontos da largura da imagem
- $b(n)$ Resposta impulsiva do filtro $H_{1/60}$
- $y_b(n)$ Sinal resultante da aplicação do filtro $H_{1/60}$
- $y_{ab}(n)$ Sinal resultante da subtração do sinal $y_b(n)$ ao sinal $y_a(n)$
- th_i Limiar para a linha i
- $Th(I(i, j), y_a(n), y_b(n))$ Limiarização da imagem $I(i, j)$, por comparação de $y_{ab}(n)$ com th_i
- $G^n(i, j)$ Imagem resultante da n -ésima aplicação do algoritmo de segmentação

Abreviaturas

- ADTree *Alternating Decision Tree*
- API *Application Programming Interface*
- BFTree *Best-First Decision Tree*
- ID3 *Iterative Dichotomiser 3*
- JAVA Linguagem de programação orientada a objectos
- HSB Sistema de cores formado pelas componentes (Matiz,Saturação,Brilho) do inglês (*Hue,Saturation,Brightness*)
- HSV Sistema de cores formado pelas componentes (Matiz,Saturação,Valor) do inglês (*Hue,Saturation,Value*)
- HTML *HyperText Markup Language*
- MATLAB *MATrix LABoratory*
- PICC Percentagem de Instâncias Correctamente Classificadas
- RGB Sistema de cores formado pelas componentes (Vermelho,Verde,Azul) do inglês (*Red,Green,Blue*)
- SLIT Sistema Linear e Invariante no Tempo
- SMO *Sequential Minimal Optimization*
- SVM *Support Vector Machines*
- TH Transformada de Hough
- XML *Extensible Markup Language*
- WEKA *Waikato Environment for Knowledge Analysis*

Capítulo 1

Introdução

"Food safety is always an emotive issue, although most of us probably don't give it much thought when we are actually eating."

Mark Graves and Bruce Batchelor [40]

1.1 Motivação

O controlo de qualidade de produtos certificados exige usualmente a avaliação por painéis de provadores. Um dos objectivos deste trabalho é colocar à disposição de um painel de provadores um instrumento que extraia medidas rigorosas e objectivas de um produto certificado. Este trabalho concentra-se na avaliação do queijo de Évora.

O queijo de Évora obedece a critérios bem definidos para a apreciação das suas características organolépticas, mediante diversos critérios. Alguns desses critérios podem ser analisados visualmente:

- Apreciação Externa:
 - Cilindro baixo, com abaulamento lateral e um pouco na face superior, sem bordos definidos;
 - Crosta inteira, bem formada, lisa ou ligeiramente rugosa, amarelada, que escurece quando em contacto com o ar, consistência dura ou semidura.
- Apreciação Interna:
 - Pasta fechada e bem ligada, com aspecto untuoso e com alguns olhos pequenos, cor amarelada e uniforme.

Em [14] encontra-se a seguinte definição para o queijo de Évora:

Definição - Queijo curado de pasta dura ou semi-dura com poucos ou nenhuns olhos ligeiramente amarelada. Obtido por esgotamento da coalha após a coagulação do leite cru de ovelha estreme, por acção de uma infusão de cardo (*Cimara Cardunculos L.*), que lhe confere um sabor ligeiramente picante e acidulado, sendo mais acentuado no queijo de pasta dura. Obrigatoriamente proveniente da região proposta e submetido ao controle da Entidade Certificadora.

Relativamente às características e forma do queijo de Évora, estas são apresentadas em [14] do seguinte modo:

Características - É um queijo curado de pasta dura, com teor de humidade entre 49% e 56% referido ao queijo isento de matéria gorda e com um teor de gordura de 45% a 60% referido ao extracto seco, sendo por isso considerado um queijo gordo; ou queijo curado de pasta semi-dura, com um teor de humidade de 54% a 63%, referido ao queijo isento de matéria gorda e com um teor de gordura de 45% a 60%, no extracto seco sendo igualmente um queijo gordo.

Forma - Cilindro baixo (prato), com abaulamento lateral e um pouco na face superior, sem bordos definidos.

Quanto às dimensões o diâmetro deve estar compreendido entre 4 e 8cm, altura entre os 2 a 4cm e peso de 60 a 90g [14].

Os critérios de apreciação externa e interna podem ser reconhecidos e analisados pelo processamento de fotografias digitais. Já os critérios de cheiro e sabor, são impossíveis de avaliar através de imagem.

Todos os produtores certificados têm de periodicamente submeter alguns queijos a uma Entidade Certificadora que, por sua vez, os envia para o painel de provadores do Laboratório de Tecnologia e Qualidade dos Produtos Naturais, no Pólo da Mitra da Universidade de Évora, que emite o seu parecer para a Entidade Certificadora.

Embora exista uma definição formal das características do queijo de Évora, usualmente não existe consenso relativamente à classificação e avaliação de um queijo por parte de todos os elementos do painel. A utilização de ferramentas com medidas rigorosas e objectivas, poderão potencialmente permitir uma avaliação mais objectiva e consistente por parte do painel de provadores.

Após cuidada pesquisa na mais diversa bibliografia especializada tanto no domínio de técnicas de processamento de imagem, como de análise de produtos naturais com as referidas técnicas, concluiu-se que se estava perante um problema ainda por explorar.

Posto isto, a abordagem seguida neste trabalho teve em consideração a complexidade crescente das soluções desenvolvidas, em função dos resultados e problemas encontrados.

1.2 Contributos

As principais contribuições desta tese são:

- Algoritmo de Segmentação, que permite o reconhecimento dos olhos existentes no queijo, com cor de fundo variável, mediante o processamento de fotografias digitais;
- Aplicação da transformada de Hough para a detecção de textura em fotografias do exterior de queijos;
- Desenvolvimento de uma aplicação de software, que permite a avaliação automática de queijos;
- Teste do software desenvolvido com fotografias e dados reais, obtidos com a colaboração do painel de provadores do Laboratório de Tecnologia e Qualidade dos Produtos Naturais, da Universidade de Évora.

1.3 Estrutura da tese

A tese encontra-se estruturada do seguinte modo:

Capítulo 1 - Capítulo introdutório da tese, onde é apresentado o problema a solucionar. É descrita a estrutura da tese e as suas principais contribuições.

Capítulo 2 - É efectuada uma revisão da literatura e estado da arte.

Capítulo 3 - Resume alguns conceitos essenciais sobre Sinais, Sistemas e Filtros Digitais.

Capítulo 4 - Descrição e apresentação do modo como são adquiridas e pré-processadas as fotografias.

Capítulo 5 - Formulação teórica do algoritmo de segmentação desenvolvido e implementado. Algumas considerações sobre o algoritmo desenvolvido e comparação com algoritmos já referenciados na literatura. Análise dos resultados da segmentação conseguidos através da sua aplicação às imagens do *corpus*.

Capítulo 6 - Apresentação de uma metodologia para a detecção de texturas, baseada na transformada de Hough. Comparação com outras abordagens.

Capítulo 7 - Descrição do procedimento de recolha de características das imagens, com base no resultado do algoritmo de segmentação apresentado no capítulo 5, assim como as características referentes à cor.

Capítulo 8 - Aplicação de algoritmos de aprendizagem automática. Descrição dos algoritmos utilizados. Análise dos resultados de cada algoritmo.

Capítulo 9 - Apresentação de uma aplicação de software, usando os algoritmos apresentados nos capítulos anteriores. É descrito o seu modo de funcionamento e interacção com o utilizador. São ainda detalhados alguns aspectos relevantes da sua implementação.

Capítulo 10 - Por fim, são apresentadas as principais conclusões, deixando-se em aberto problemas para investigação futura.

Capítulo 2

Estado da Arte

"Image processing and analysis are specialist subjects, and their application is complicated by the sheer volume of data that as to be processed ... Because of the specialist nature of image processing on the one hand and the demands of food production on the other, this is no easy task."

E. R. Davies [20]

O processamento de imagem é uma técnica que tem sido amplamente utilizada para a análise de diversos materiais, produtos naturais, processos de produção, entre outros, com o intuito de detectar defeitos, melhorar a eficiência e a qualidade de produção [40].

A avaliação e análise de produtos naturais por meio do processamento de imagem, tem sido efectuado por diversos autores com os mais diversos objectivos, dos quais se destacam o controlo de qualidade, selecção ou medição de características e detecção de anomalias.

Propriedades como a cor, forma, tamanho ou manchas na superfície são aspectos que o consumidor tem em consideração aquando da compra de fruta. Para facilitar a classificação e embalagem de fruta comercializada nas grandes superfícies comerciais, foram desenvolvidas tecnologias que por análise de imagens permitem automatizar esta tarefa [40].

Ainda em [40], o processamento de imagens digitais é utilizado para a classificação de superfícies de madeira. Pretende-se detectar rachas, nós e buracos de vermes, influenciando na classificação da madeira. Este processamento é efectuado mediante a utilização de técnicas de detecção de textura, como por exemplo, os filtros de Gabor [21].

Para os criadores de animais, um dado importante é a possibilidade de monitorizar o tamanho e peso dos seus animais. A recolha de imagens pode requerer sistemas complexos, nomeadamente para imagens de peixes. No entanto, obtida a imagem prossegue-se com a aplicação de técnicas de processamento de imagem [17]. O cálculo do peso de porcos, através do processamento de imagem, tem sido igualmente objecto de diversos estudos [2], [58].

O estudo da qualidade dos cereais, tem também sido alvo de técnicas de processamento de imagem, permitindo a automatização deste processo [1], [33].

Na maioria dos exemplos referidos, é necessário o reconhecimento de objectos na imagem. A segmentação de imagens, método pelo qual uma imagem é dividida nas regiões que a constituem, é bastante utilizado para a detecção de objectos em imagens. Em [53], os algoritmos de segmentação são divididos em duas categorias: descontinuidade ou similaridade. Na primeira categoria, a imagem é dividida em função de alterações bruscas da sua cor, correspondendo aos limites (fronteiras) dos objectos, descrito na literatura como *edge detection* [51]. Na segunda categoria, é definido um critério *a priori*, sendo a imagem particionada em regiões que estão de acordo com o critério. Entre outras técnicas, a limiarização (*thresholding*) apresenta-se como um dos métodos mais utilizados na segmentação, devido à sua simplicidade de implementação [53].

A segmentação de uma imagem por limiarização, consiste numa forma de separar os valores dos pixels da imagem em duas classes distintas, em função do nível do limiar (*threshold*) pretendido. Como resultado, é gerada uma imagem binária onde os pixels correspondentes aos objectos assumem o valor 1 e o fundo da imagem assume o valor 0. Para tal define-se a função de limiarização pela equação 2.1:

$$F_T(I(i, j), \tau) = \begin{cases} 1 & \text{se } I(i, j) \geq \tau \\ 0 & \text{se } I(i, j) < \tau \end{cases} \quad (2.1)$$

Diversos métodos utilizam a limiarização como técnica de segmentação de imagens como, por exemplo, Otsu Threshold [48], sendo uma técnica referenciada por diversos autores [31], [62], [74].

Na maioria dos algoritmos de segmentação por limiarização é definido um limiar fixo por avaliação do histograma da imagem. Uma abordagem diferente, consiste em considerar a definição automática do limiar, segundo a intensidade de cor e tamanho dos objectos, bem como, a fracção da área ocupada por estes, entre outras características obtidas pela análise do histograma [35]. São diversos os métodos para o cálculo do limiar de forma automática, dos quais se destacam *iterative thresholding* [25], *adaptive thresholding* [23], [75] e *variable thresholding* [35].

Outros métodos de segmentação, como por exemplo, segmentação por *clustering* [3], [28], segmentação baseada em regiões (*region-based segmentation*) [35], segmentação por reconhecimento das fronteiras dos objectos (*boundary segmentation*) [34], são igualmente referenciados na literatura.

Outro ponto de vista importante para esta tese é o tratamento de dados/informação obtidos pelo processamento da imagem. Este processamento pode gerar elevadas quantidades de dados.

As elevadas quantidades de dados resultantes da introdução das tecnologias de informação e comunicação em praticamente todas as actividades do quotidiano, tem levado ao desenvolvimento de algoritmos para a extracção de informações e conhecimentos úteis a partir dos dados em bruto [43]. Esta necessidade de extrair informação de forma automática de bases de dados, conduziu ao conceito de mineração de dados [18], [37].

A descoberta de conhecimento em grandes quantidades de dados, requer a definição de um modelo, ou seja, um sumário global do conjunto de dados. Este modelo pode tomar várias formas tais como árvores, grafos, regras ou equações [43].

Para o problema em análise nesta tese - a classificação automática do queijo de Évora - pretende-se a definição de um modelo que permita prever a classificação de um queijo, mediante o conhecimento prévio da classificação de outros queijos (base de conhecimento). No domínio da mineração de dados, este problema é denominado por predição, que pode ser de dois tipos:

Classificação Atribuição de uma classe em função dos atributos (variáveis de entrada) de uma instância.

Regressão Aproximação de uma função desconhecida com um contradomínio numérico.

Existem diversas ferramentas de software de mineração de dados, tais como Weka¹, MATLAB², KDnuggets³. A ferramenta Weka, é uma das mais utilizadas, e estando desenvolvida na linguagem de programação JAVA, constituiu-se como a ferramenta de mineração de dados preferencial para a integração com o software desenvolvido no âmbito desta tese.

Para problemas de classificação, os algoritmos baseados em árvores de decisão são os mais utilizados, sendo o algoritmo C4.5 desenvolvido por Quinlan, um dos mais populares [73]. Além do algoritmo C4.5, foram desenvolvidos outros algoritmos baseados em árvores de decisão, tais como ADTree (*Alternating Decision Tree*) [24], BFTree (*Best-First Decision Tree*) [60], *Random Forest* [72].

Além dos algoritmos referidos no parágrafo anterior, as SVM (*Support Vector Machines*) desenvolvidas em [16], são também utilizadas em problemas de classificação, estando igualmente disponíveis no Weka.

¹<http://www.cs.waikato.ac.nz/ml/weka/>

²<http://www.mathworks.com/products/matlab/>

³<http://www.kdnuggets.com/index.html>

Capítulo 3

Sinais, Sistemas e Filtros Digitais

"Signal processing is concerned with the representation, transformation, and manipulation of signals and the information they contain. For example, we may wish to separate two or more signals that have somehow been combined, or we may want to enhance some signal component or some parameter of a signal model."

Alan V. Oppenheim, Ronald W. Schaffer [5].

3.1 Introdução

No presente capítulo é efectuada uma breve descrição dos conceitos principais utilizados ao longo do trabalho.

3.2 Sinais digitais

Os sinais estão presentes em muitos domínios da ciência e da engenharia, como por exemplo, nas comunicações, na biologia, na sismologia, na electrónica, no processamento de imagem ou de fala, nos sistemas militares e na indústria, entre muitos outros [59], [7].

Segundo [7], um sinal é definido como uma quantidade, propriedade ou variável que depende do tempo. Em [39] é apresentada uma definição mais formal, em que um sinal é definido como uma função de uma ou mais variáveis independentes com informação sobre determinado fenómeno físico.

Como exemplo de sinais poderemos considerar a variação da força de um sismo ao longo do tempo, figura 3.1a, ou a variação do brilho numa imagem ao longo de duas variáveis espaciais x e y , que definem o plano da imagem, figura 3.1b.

Tendo em consideração o domínio de um sinal, este pode ser contínuo ou discreto. Um sinal contínuo está definido para todos os valores da variável independente (e.g. tempo ou espaço). Um sinal discreto encontra-se apenas definido em determinados valores da variável independente, eventualmente espaçados regularmente.

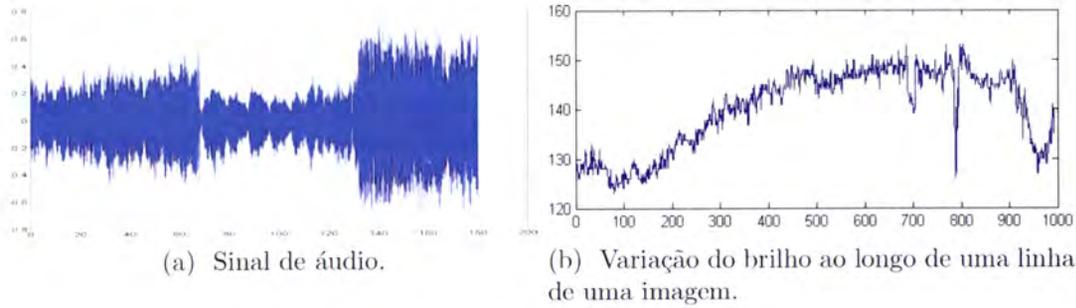


Figura 3.1: Exemplo de sinais.

A análise de sinais contínuos não será objecto de estudo do presente texto, visto que o tempo discreto intrínseco ao funcionamento de um computador inviabiliza o desenvolvimento de técnicas para processamento de sinais contínuos.

Qualquer sinal discreto, $x(n)$, pode ser representado em função de um sinal impulso unitário, também designado por impulso unitário de Kronecker, representado na figura 3.2.



Figura 3.2: Sinal impulso unitário de Kronecker.

O sinal impulso unitário, que se denota $\delta(n)$, é definido por

$$\delta(n) = \begin{cases} 0 & ; \quad n \neq 0 \\ 1 & ; \quad n = 0 \end{cases} \quad (3.1)$$

permitindo a decomposição de qualquer sinal discreto $x(n)$, numa soma ponderada e deslocada de impulsos unitários, da seguinte forma:

$$x(n) = \dots + x(-1)\delta(n+1) + x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \dots \quad (3.2)$$

A expressão anterior, poderá ser redefinida para uma forma condensada, consistindo num somatório em que $x(k)\delta(n-k)$ é um sinal com a amplitude de $x(k)$ no instante em que $n = k$, sendo que para os restantes instantes de n , a amplitude de $x(k)$ é igual a zero.

$$x(n) = \sum_{k=-\infty}^{+\infty} x(k)\delta(n-k) \quad (3.3)$$

Para um sinal discreto a duas dimensões, a expressão do sinal impulso unitário da equação 3.1, é redefinida por

$$\delta(u, v) = \begin{cases} 1 & ; \quad u = v = 0 \\ 0 & ; \quad u \neq 0 \text{ e } v \neq 0 \end{cases} \quad (3.4)$$

Interpretando a equação 3.1 no âmbito de uma imagem digital, esta representa um pixel brilhante (com valor igual a 1) na coordenada da origem de um plano escuro e infinito.



Figura 3.3: Representação do impulso de Dirac a duas dimensões $\delta(u, v)$.

3.3 Sistemas digitais

Um sistema é um processo que estabelece uma relação de causa-efeito entre o sinal de entrada $x(n)$, e o sinal de saída $y(n)$. Um sistema pode ser igualmente definido como um dispositivo físico que realiza uma operação ou modificação sobre um sinal [32]. Uma utilização comum de um sistema é a redução de ruído ou interferência presente num sinal, que corrompe a informação contida no sinal. Esta operação, também designada de filtro, reduz o ruído (filtra) presente no mesmo.

Quando um sinal passa por um sistema, em que é filtrado e por isso é modificado, está-se perante uma operação de processamento de sinal. Os sistemas são, na sua generalidade, classificados consoante o tipo de operação que aplicam a um sinal. Se a operação for linear, o sistema é igualmente linear. Se a operação aplicada sobre o sinal for não linear, o sistema é designado por sistema não linear. Neste trabalho serão utilizados maioritariamente sistemas lineares.

Os sistemas além de caracterizados segundo as operações que realizam sobre um sinal, podem ser caracterizados segundo o tipo do domínio dos sinais que processam. Em [39], um sistema é referido como contínuo ou discreto se transforma sinais de entrada contínuos ou discretos em sinais de saída contínuos ou discretos, respectivamente.

Tendo em consideração algumas propriedades dos sistemas, estes podem ser caracterizados por apresentarem ou não apresentarem memória¹, por serem causais ou não causais².

Os sistemas são divididos em dois grandes grupos: os sistemas lineares e os sistemas não lineares. Se um sistema que recebe os sinais de entrada $x_1(n) + x_2(n)$ produz o sinal de saída $y_1(n) + y_2(n)$, então o sistema satisfaz a condição de aditividade. Se o sinal de saída $\beta y_1(n)$ corresponde ao sinal de entrada $\beta x_1(n)$, em que β é uma constante arbitrária, então o sistema satisfaz a condição de homogeneidade. Um sistema que satisfaça as condições de homogeneidade e de aditividade é designado por sistema linear. Um sistema não linear é um sistema que não satisfaz alguma das condições anteriores [59].

As condições para que um sistema seja linear, são resumidas pelas próximas expressões:

$$\text{Aditividade} : x_1(n) + x_2(n) \longrightarrow y_1(n) + y_2(n) \quad (3.5)$$

$$\text{Homogeneidade} : \beta x_1 \longrightarrow \beta y_1 \quad (3.6)$$

As duas propriedades anteriores, podem ser combinadas, resultando na propriedade de sobreposição:

$$\beta_1 x_1(n) + \beta_2 x_2(n) \longrightarrow \beta_1 y_1(n) + \beta_2 y_2(n) \quad (3.7)$$

Outra propriedade importante na caracterização de um sistema, refere-se à invariância temporal do sistema, ou seja, um sistema é considerado invariante no tempo se uma translação de N instantes no tempo do sinal de entrada t provoca igualmente uma translação no sinal de saída $t - N$.

Um sistema é considerado linear e invariante no tempo, SLIT, se verificar as condições de linearidade e invariância no tempo. Então, se se conhecer a resposta de um SLIT a um sinal impulso unitário, é possível determinar a resposta do SLIT a qualquer sinal de entrada, que conduz ao conceito de resposta impulsiva do sistema.

A resposta impulsiva, $h(n)$, também designada por resposta impulsional [39], define o sinal de saída de um SLIT cujo sinal de entrada é um impulso unitário de Dirac. A forma de determinar a resposta de um SLIT a qualquer sinal de entrada, é dada pela soma de convolução, assunto abordado na próxima secção.

3.4 Convolução

Tal como referido anteriormente, o conceito de soma de convolução, ou simplesmente convolução, permite determinar a relação entre o sinal de entrada de um sistema linear e invari-

¹Num sistema sem memória a saída no instante t depende apenas da entrada no instante t , ao contrário de um sistema com memória em que a saída no instante t depende não só da entrada no instante t como de entradas anteriores.

²Num sistema causal a saída no instante t pode depender não só da entrada no instante t como de entradas passadas, sendo que os valores de saída de um sistema não causal dependem de valores futuros da entrada.

ante no tempo, $x(n)$, e o sinal de saída, $y(n)$.

Segundo a equação 3.3, qualquer sinal discreto pode ser representado pela soma de impulsos unitários e tendo em consideração as propriedades de linearidade e invariância temporal de um SLIT, tem-se que:

$$\delta(n) \longrightarrow h(n) \implies \delta(n - k) \longrightarrow h(n - k) \quad (3.8)$$

$$\begin{aligned} \delta_k(n) = \delta(k - n) &\longrightarrow h_k(n) = h(n - k) \implies \\ \implies x(n) = \sum_{k=-\infty}^{+\infty} a_k \delta_k(n) &\longrightarrow y(n) = \sum_{k=-\infty}^{+\infty} a_k h_k(n) \end{aligned} \quad (3.9)$$

então, fazendo $a_k = x(k)$, torna-se possível determinar a resposta de um SLIT, pela operação de convolução dada pela equação 3.9.

$$y(n) = \sum_{k=-\infty}^{+\infty} x(k)h(n - k) = x(n) \circledast h(n) \quad (3.10)$$

O operador de convolução, \circledast , é um operador linear³, pelo que goza de propriedades importantes, incluindo as propriedades comutativa, associativa e distributiva.

1. **Comutativa:** $x(n) \circledast h(n) = h(n) \circledast x(n)$
2. **Associativa:** $x(n) \circledast [h_1(n) * h_2(n)] = [x(n) * h_1(n)] \circledast h_2(n)$
3. **Distributiva:** $x(n) \circledast [h_1(n) + h_2(n)] = x(n) \circledast h_1(n) + x(n) \circledast h_2(n)$

3.5 Filtros digitais

Os sinais digitais, na sua generalidade, contêm bastante informação sobre o fenómeno físico que representam, embora nem toda a informação seja substancialmente útil. A eliminação dessas características, que por vezes estão associadas a ruído⁴, é realizada pela aplicação de um filtro ao sinal. Com esta operação, é produzido um novo sinal resultante de uma alteração do sinal original [66].

³Um operador L é linear, se para qualquer par de funções f, g e o escalar t , verifica:

- $L(f + g) = L(f) + L(g)$
- $L(tf) = tL(f)$

⁴O ruído pode ser definido como um sinal indesejável que interfere com a comunicação, medição ou processamento da informação contida num sinal [67].

Os filtros digitais podem ser classificados em duas classes, ou seja, em filtros lineares e não lineares, dependendo das propriedades matemáticas da função que implementa o filtro.

Os filtros lineares combinam os valores dos pixels da imagem a filtrar de uma forma linear, através de somas ponderadas. Por outras palavras, o cálculo do novo valor do pixel na imagem filtrada, $I'(x, y) = p_{i,j}$, é obtido segundo a expressão

$$I'(x, y) = \sum_{(i,j) \in R_H} I(x+i, y+j) \cdot H(i, j) \quad (3.11)$$

onde H é uma matriz $n \times n$, designada por «máscara do filtro», em que cada elemento $H(i, j)$ indica o peso do pixel correspondente na região de suporte, na soma da equação 3.11. A equação 3.11 é uma aplicação do conceito de soma de convolução, descrito na secção 3.4.

A região de suporte é dada por uma matriz definida em função do tamanho do filtro, n , que para o caso de $n = 3$, a região de suporte é composta pelo pixel central $p_{i,j}$ e pelos seus 8 pixels vizinhos, tomando a forma de

$$R = \begin{bmatrix} (i-1, j-1) & (i, j-1) & (i+1, j-1) \\ (i-1, j) & (i, j) & (i+1, j) \\ (i-1, j+1) & (i, j+1) & (i+1, j+1) \end{bmatrix} \quad (3.12)$$

A aplicação do filtro a uma imagem, pode ser dividida em três passos:

1. A máscara do filtro, H , é movida sobre a imagem original $I(x, y)$, sendo que $H(0, 0)$ corresponde à posição actual na imagem (i, j) ;
2. Multiplicar todos os coeficientes de $H(i, j)$ pelo elemento correspondente da região de suporte, e somar os resultados;
3. A soma final será colocada na posição actual na nova imagem $I'(i, j)$.

Capítulo 4

Aquisição e Pré-processamento das Imagens

"Imagem enhancement is an important step in the processing of large data sets to make the results more suitable for classification than were the original data. It accentuates and sharpens the image features, such as edges, boundaries, and contrast. The process does not increase the inherent information content in the data, but it does increase the dynamic range of the features."

Sing-Tze Bow [9].

4.1 Introdução

A metodologia e os algoritmos desenvolvidos para a avaliação do queijo de Évora, através de análise e processamento de fotografias digitais, pressupõe que as fotografias sejam obtidas respeitando alguns requisitos. Com estes pressupostos pretende-se apenas definir o ambiente em que a fotografia deve ser adquirida, restringindo a que na imagem apareçam apenas elementos essenciais a um processamento adequado da imagem.

4.2 Definição do ambiente

De modo a que seja possível a avaliação de um queijo, através do processamento de uma imagem, é fundamental o reconhecimento do queijo na imagem. Outro aspecto a ter em consideração para a definição do ambiente em que as imagens são recolhidas, é que uma das características importantes na avaliação de um queijo, é o seu tamanho. Portanto, a aquisição de imagens que permitam o reconhecimento do queijo na imagem, assim como um cálculo aproximado da sua dimensão real, deverá respeitar as seguintes condições:

1. Num fundo branco ou cinzento, definir um quadrado (moldura) de 15 cm de lado, com uma cor escura de forte contraste com o fundo;
2. O fundo deve ter cor suave e uniforme, não sendo necessariamente igual dentro e fora da moldura;
3. Cortar uma fatia de queijo com cerca de 2 a 4 mm de espessura;
4. A fatia cortada deve ser colocada com a maior dimensão paralela a um dos lados da moldura e sensivelmente no centro;
5. As dimensões da moldura devem ser superiores a $2/3$ da fotografia.

As fotografias da figura 4.1, embora nem todas estejam totalmente correctas, são no entanto aceites e processáveis pela aplicação.



Figura 4.1: Exemplos de imagens aceites pela aplicação.

Nas fotografias 4.1b e 4.1c a moldura encontra-se com uma excessiva rotação, relativamente à moldura da fotografia 4.1a, que servirá como padrão preferencial para um melhor desempenho do processamento da fotografia. Não obstante, a rotação da moldura das fotografias 4.1b e 4.1c poderá ser corrigida, problema que será abordado na secção 4.5.

Os exemplos presentes na figura 4.2, são considerados incorrectos. Tendo em conta os pressupostos enunciados anteriormente, ao executar a aplicação sobre fotografias com ambiente incorrecto, o resultado poderá ser algo completamente imprevisível, não sendo responsabilidade da aplicação a correcção de todos os erros cometidos pelo utilizador na definição do ambiente, tal como ele é proposto.

4.3 Detecção da moldura

Quando é solicitado à aplicação que processe uma fotografia, por forma a avaliar um queijo, é necessário num estado inicial o reconhecimento do queijo na imagem. Se forem respeitados

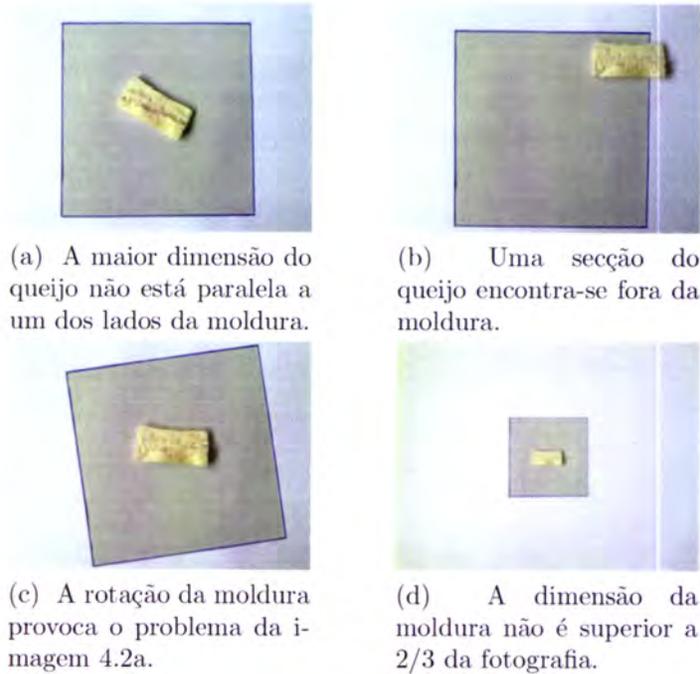


Figura 4.2: Exemplos de imagens incorrectas.

os pressupostos da definição do ambiente propostos no ponto anterior, então o queijo está algures dentro da moldura. Portanto, o primeiro processamento a aplicar sobre a imagem é a detecção da moldura.

O processo de detecção de objectos em imagens não é simples de realizar de forma automática [52], pelo que deve iniciar-se o processamento da imagem utilizando algoritmos de detecção de contornos, como por exemplo, os operadores de Roberts [55], operadores de Sobel [45], operadores de Canny [13] [12] ou operadores de Prewitt [69].

Os operadores de Prewitt, também designados por filtro de Prewitt, efectuem o cálculo do gradiente¹ da luminosidade de cada pixel da imagem, informando sobre como varia a luminosidade ao longo da imagem. As variações bruscas de cor, normalmente correspondem aos limites de objectos, pelo que os contornos dos objectos são identificados por uma nítida alteração de cor ao longo de determinada orientação.

Em termos matemáticos, o cálculo dos operadores de Prewitt recorre a filtros lineares, que utilizam duas matrizes 3×3 com o intuito de neutralizar a sensibilidade ao ruído por parte

¹O vector gradiente para uma imagem $I(u, v)$, é dado por:

$$\nabla I(u, v) = \begin{bmatrix} \frac{\partial I}{\partial u}(u, v) \\ \frac{\partial I}{\partial v}(u, v) \end{bmatrix} \quad (4.1)$$

do operador de gradiente simples. As duas matrizes, 4.2, são convolvidas com a imagem original, $I(i, j)$ por forma a calcular as variações de cor, ao longo da direcção horizontal e vertical.

$$(a) \quad H_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{e} \quad (b) \quad H_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.2)$$

A magnitude do vector gradiente é dada pela equação 4.3. A magnitude do vector gradiente é invariante à rotação da imagem e, como tal, independente da orientação dos objectos presentes na imagem.

$$\begin{aligned} |\nabla I|(i, j) &= \sqrt{\left(\frac{\partial I}{\partial i}(i, j)\right)^2 + \left(\frac{\partial I}{\partial j}(i, j)\right)^2} \\ |\nabla I|(i, j) &= \sqrt{H_x^2 + H_y^2} \end{aligned} \quad (4.3)$$

Visto que a abordagem desenvolvida efectua o processamento das imagens apenas a uma dimensão, pretende-se apenas detectar as linhas verticais da moldura. Para tal foram utilizados os operadores de Prewitt para detecção de linhas verticais, que podem ser implementados recorrendo-se à operação de convolução linear, dada pela equação 4.4.

$$\begin{aligned} Prewitt(i, j) &= \sum_{n=-1}^{n=1} \sum_{m=-1}^{m=1} h_{linha}(i, j) I(i-n, j-m) \\ &= h_{linha}(i, j) \otimes I(i, j) \end{aligned} \quad (4.4)$$

onde $h_{linha}(i, j)$ é uma matriz 3×3 que define a resposta impulsiva, dada pela equação 4.2a.

Os valores do gradiente são então estimados, pela equação 4.5, numa escala apropriada.

$$\nabla I(i, j) \approx \frac{1}{6} \cdot \begin{bmatrix} (I * H_x)(i, j) \\ (I * H_y)(i, j) \end{bmatrix} \quad (4.5)$$

Após a filtragem da imagem com os operadores de Prewitt, a imagem resultante contém algum ruído. Este ruído apresenta-se na forma de pixels isolados de tonalidade acinzentada, o que pode dificultar a localização efectiva da moldura. A forma de eliminação deste ruído é apresentado na próxima secção.

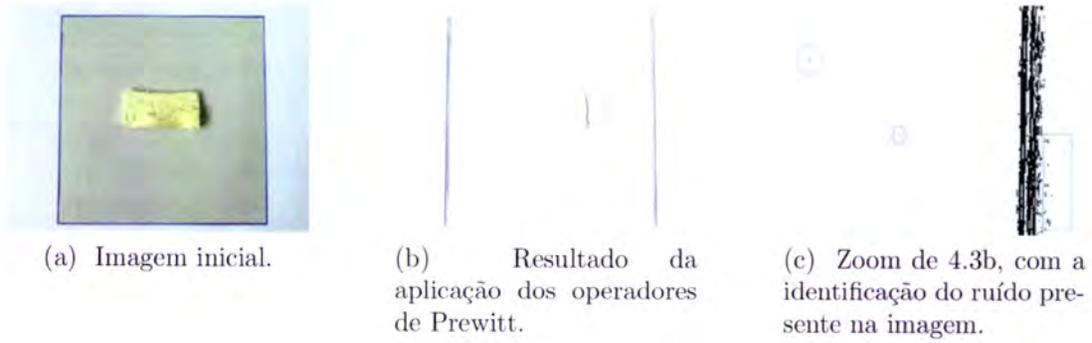


Figura 4.3: Detecção da moldura com os operadores de Prewitt.

4.4 Eliminação de ruído

A eliminação do ruído é efectuada pela filtragem da imagem $I(i, j)$ com um filtro de mediana. Sendo impossível conceber um filtro que elimine todo o ruído sem afectar as estruturas importantes da imagem, o filtro de mediana constitui uma boa alternativa [71].

Dada a matriz I com i colunas e j linhas, e dois inteiros positivos ímpares, u e v , o filtro de mediana calcula uma matriz *Mediana* com o mesmo tamanho de I , em que $Mediana(i, j)$ contém a mediana dos números em $I(i, j)$, correspondendo à vizinhança de $u \times v$ em torno de (i, j) . Quando a vizinhança de uma coordenada (i, j) não estiver definida, convencionase que $Mediana(i, j) = 0$.

A figura 4.4 demonstra a computação do filtro de mediana para valores de u e v ambos iguais a 3.

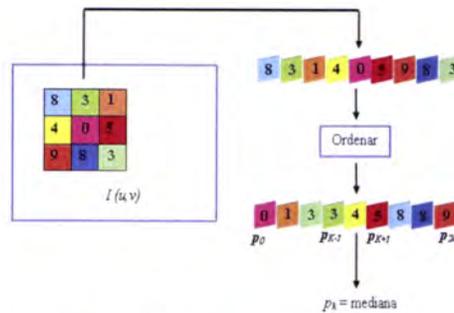


Figura 4.4: Esquema da aplicação do filtro de mediana.

A mediana de um conjunto de valores p_i é definida por:

$$mediana(p_0, p_1, \dots, p_K, \dots, p_{2K}) \triangleq p_K \tag{4.6}$$

em que o valor da mediana corresponde ao valor central p_K , se a sequência $(p_0, p_1, \dots, p_K, \dots, p_{2K})$ estiver ordenada, ou seja, se para qualquer $i : p_i \leq p_{i+1}$.

O resultado da aplicação do filtro de mediana à figura 4.3c, resulta na figura 4.5b.

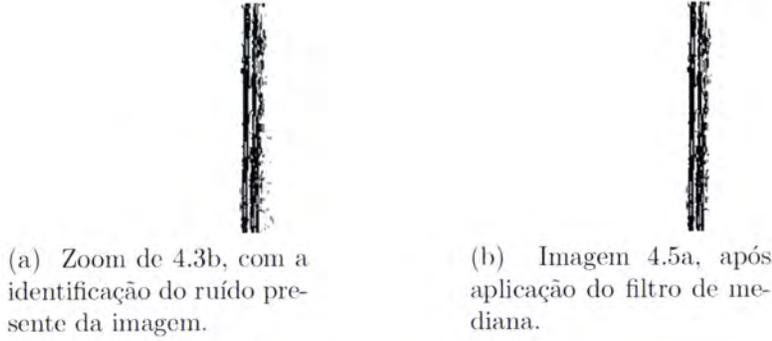


Figura 4.5: Aplicação do filtro de mediana.

4.5 Rotação da imagem

Identificadas as linhas verticais da moldura, prossegue-se verificando se a moldura apresenta algum grau de inclinação. Se existir inclinação da moldura esta deve ser removida, de modo a não influenciar no cálculo da largura da moldura, e por conseguinte, afectar o cálculo da dimensão real do queijo.

Para o cálculo do ângulo de inclinação da moldura, são determinados dois pontos da linha esquerda da moldura, que distam $1/3$ da altura da imagem. Para ambos os pontos, (x_1s_i, y_1s_1) e (x_2s_i, y_2s_1) , é amostrado um conjunto de valores num intervalo de 5% abaixo de y_1s_1 e y_2s_1 , de modo a estimar o valor de x_1s_i e x_2s_i . Este conjunto de valores é definido de acordo com:

$$(y_1s_1, \dots, y_1s_n), \quad y_1s_1 = 1/3 \times j \quad \text{e} \quad y_1s_n = y_1s_1 + (j \times 0.05) \quad (4.7)$$

$$(y_2s_1, \dots, y_2s_n), \quad y_2s_1 = 2/3 \times j \quad \text{e} \quad y_2s_n = y_2s_1 + (j \times 0.05) \quad (4.8)$$

que resulta em dois conjuntos de pontos:

- $[(x_1s_1, y_1s_1), (x_1s_2, y_1s_2), \dots, (x_1s_n, y_1s_n)]$
- $[(x_2s_1, y_2s_1), (x_2s_2, y_2s_2), \dots, (x_2s_n, y_2s_n)]$

O valor da coordenada x_1s_i do ponto (x_1s_i, y_1s_1) , é calculada pela média dos três maiores valores do conjunto de coordenadas $[x_1s_1, x_1s_2, \dots, x_1s_n]$, sendo efectuado o mesmo procedimento para o cálculo da coordenada x_2s_i do ponto (x_2s_i, y_2s_1) , figura 4.6. Este procedimento permite evitar a influência de algum ruído que possa persistir mesmo após a aplicação do filtro de mediana.

O ângulo de rotação, θ , é então calculado pela equação 4.9

$$\theta = \arctan \left(\frac{x_2s_i - x_1s_i}{y_2s_1 - y_1s_1} \right) \quad (4.9)$$

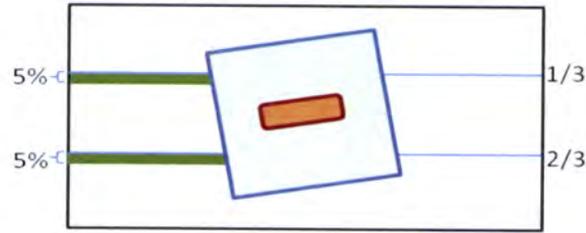


Figura 4.6: Determinação dos pontos para o cálculo do ângulo de rotação.

Se o ângulo de rotação θ , for superior a um grau ($0.017453rad$), será aplicado sobre a imagem o algoritmo apresentado em 1. Caso contrário, não será realizado nenhum processamento sobre a imagem.

O pseudo-código seguinte, apresenta o algoritmo que permite a rotação da imagem, segundo o seu centro:

Algorithm 1

```

1: procedure ROTATION( $\theta$ )
2:    $G(i, j) \leftarrow new \ Image(i, j)$   $\triangleright G$  é uma nova imagem com as mesmas dimensões da
   imagem original,  $I(i, j)$ 
3:   for all As coordenadas  $(i, j)$  de  $G$  do
4:      $a_0 = \cos \theta$ 
5:      $b_0 = \sin \theta$ 
6:      $a_1 = -b_0$ 
7:      $b_1 = a_0$ 
8:     for all  $y < j$  do
9:       for all  $x < i$  do
10:         $i' = x - (i/2)$ 
11:         $j' = y - (j/2)$ 
12:         $ri = round((a_0 \times i') + (a_1 \times j'))$ 
13:         $rj = round((b_0 \times i') + (b_1 \times j'))$ 
14:         $i' = ri + (i/2)$ 
15:         $j' = rj + (j/2)$ 
16:        if  $(ri, rj)$  está dentro de  $G$  then
17:           $G(ri, rj) = I(i, j)$ 
18:        end if
19:      end for
20:    end for
21:  end for
22: end procedure

```

4.6 Reconhecimento do queijo

O último pré-processamento a aplicar à imagem I , pretende reconhecer a zona correspondente ao queijo. A pesquisa pelo queijo é efectuada apenas dentro da moldura, pelo que é necessário a distinção entre a cor do fundo e cor do próprio queijo. No espaço de cores RGB esta distinção não é simples, visto existir uma sobreposição das regiões de cores correspondentes ao fundo da moldura e ao queijo, dificultando a separação de ambas as regiões.

Surgiu então a necessidade de proceder à transformação do espaço de cores RGB para HSV. No espaço de cores HSV, as cores são especificadas pelas componentes de matiz, saturação e valor. O espaço de cores HSV também pode ser designado por HSB em que a componente B remete-nos para o brilho da cor.

4.6.1 Conversão de RGB para HSV

Apresentam-se, de seguida, os cálculos necessários à conversão do espaço de cores RGB para o espaço de cores HSV:

$$M = \max(R, G, B) \quad (4.10)$$

$$m = \min(R, G, B) \quad (4.11)$$

$$C = M - m \quad (4.12)$$

$$H' = \begin{cases} \text{indefinido} & , \quad C = 0 \\ \frac{G-B}{C} \bmod 6 & , \quad M = R \\ \frac{B-R}{C} + 2 & , \quad M = G \\ \frac{R-G}{C} + 4 & , \quad M = B \end{cases} \quad (4.13)$$

$$H = 60^\circ \times H' \quad (4.14)$$

$$V = M \quad (4.15)$$

$$S = \begin{cases} 0 & , \quad C = 0 \\ \frac{C}{V} & , \quad c.c. \end{cases} \quad (4.16)$$

O espaço de cores RGB é tradicionalmente representado através de uma figura geométrica denominada de «cubo de cores», figura 4.7a, já o espaço de cores HSV é representado por um «hexágono de cores», figura 4.7b.

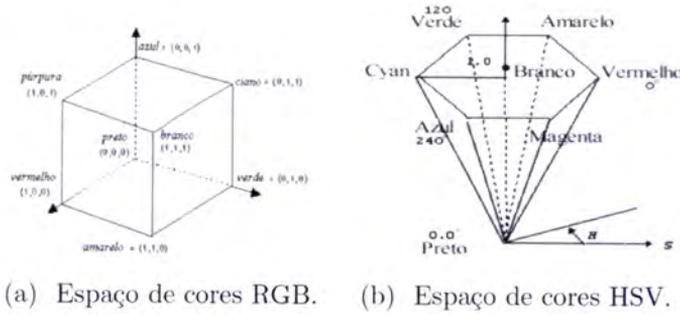


Figura 4.7: Representação dos espaços de cor RGB e HSV.

4.6.2 Separação do queijo da moldura

Foram determinados os limites de cor no espaço HSV sobre um conjunto representativo de imagens de queijos. Em HSV as cores tornam-se mais perceptíveis, visto que este espaço de cores foi projectado para imitar a forma como o olho humano percebe a cor. Pelo que, em HSV é mais intuitiva a distinção das diferentes regiões da imagem.

Na figura 4.8a encontra-se a região de cores correspondente à cor do fundo da moldura, em função das componentes H e S. Na figura 4.8b encontra-se a região de cores correspondente à moldura e ao queijo simultaneamente, também em função das componentes H e S. Conclui-se então, que a região do queijo tem valores de S superiores a 0.15, sendo esta a forma de separação entre a moldura e o queijo.

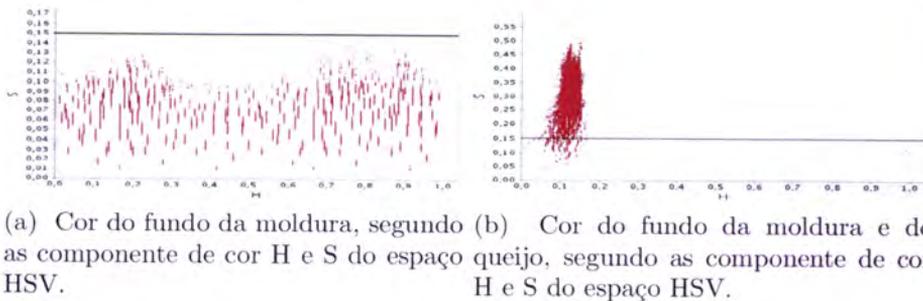


Figura 4.8: Distinção entre a região de cores do fundo da moldura e do queijo.

4.6.3 Cálculo das dimensões do queijo

Na secção 4.2 é referido que para efectuar o cálculo aproximado das dimensões do queijo, a fotografia deve ser adquirida com o queijo inserido numa moldura quadrada com 15cm de lado.

Se este pressuposto for respeitado, o cálculo das dimensões do queijo, é realizado de forma simples pelo algoritmo apresentado em 2.

Algorithm 2

```
1: procedure DIMENSOES(larguraMoldura, larguraImagem, alturaImagem) ▷  
   Os argumentos são dados em número de pixels  
2:   larguraQueijo  $\leftarrow 15 \times \text{larguraImagem} / \text{larguraMoldura}$   
3:   alturaQueijo  $\leftarrow 15 \times \text{alturaImagem} / \text{larguraMoldura}$   
4:   return (larguraQueijo, alturaQueijo)  
5: end procedure
```

Capítulo 5

Segmentação

"Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the eventual success or failure of computerized analysis procedures."

R. Gonzalez and R. Woods [53]

5.1 Introdução

No presente capítulo é apresentada, porventura, a principal contribuição desta tese, que consiste num algoritmo baseado em técnicas de processamento de sinal, para efectuar a detecção dos olhos do queijo, numa imagem digital.

5.2 Apresentação do algoritmo desenvolvido

Em termos de cor, os olhos distinguem-se do fundo amarelado do queijo, pois apresentam uma cor mais escura, notando-se uma acentuada variação de cor nas fronteiras dessas regiões. Os olhos ao apresentarem uma cor mais escura relativamente ao fundo do queijo, assumem uma cor acastanhada, o que se traduz numa diminuição do valor da componente azul do espaço de cores RGB. Portanto, a visualização da imagem de um queijo tendo apenas em consideração a componente azul do espaço RGB, permite uma melhor intuição sobre a localização dos olhos, como se pode observar na figura 5.1.

A metodologia desenvolvida para a detecção dos olhos do queijo de Évora, pressupõe um processamento horizontal da imagem, ou seja, o processamento é apenas realizado sobre as linhas da imagem, a uma dimensão. Na figura 5.2 é possível observar a variação de cor ao longo de uma linha da imagem da figura 5.1a, relativamente a cada uma das três componentes do espaço de cores RGB. Os três sinais presentes na figura 5.2 apresentam valores de variância bastante distintos: $R = 163$, $G = 576$ e $B = 953$. Sendo o sinal da



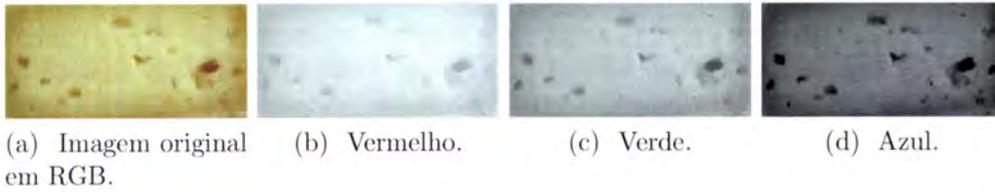


Figura 5.1: Imagem visualizada em cada uma das três componentes de cor RGB.

componente azul o que apresenta maior variância, esta é a componente de cor que permite uma melhor detecção das regiões dos olhos.

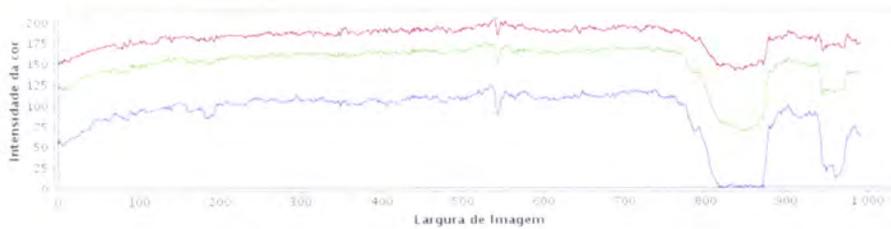


Figura 5.2: Variação de cada uma das componentes de cor para uma linha da imagem 5.1a.

Por forma a diminuir a presença de ruído e a detectar variações bruscas num sinal, foram desenhados e implementados filtros de média móvel, com características e respostas impulsivas diferentes.

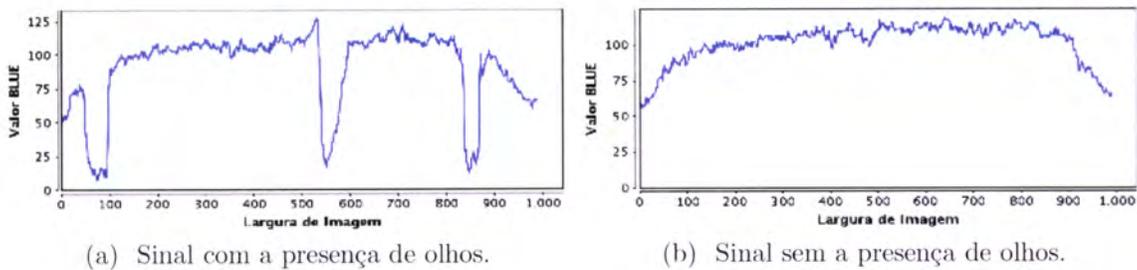


Figura 5.3: Diferença da variação de cor em sinais com e sem a presença de olhos.

O filtro de média móvel é provavelmente o filtro mais utilizado em processamento de sinal, visto ser bastante compreensível e fácil de implementar [63]. Este filtro, utilizado para a redução de ruído aleatório, realiza uma média móvel de K pontos do sinal de entrada, $x(n)$, para produzir cada ponto do sinal de saída, $y(n)$. A implementação, pela operação de convolução, resulta na equação 5.1.

$$y(n) = \frac{1}{K} \sum_{i=0}^{K-1} x(n+i) \quad (5.1)$$

O filtro de média móvel representado na equação 5.1, realiza a média dos pontos do sinal de entrada, compreendidos entre $x(n)$ e $x(n + K - 1)$. No entanto, para a implementação de um filtro de média móvel, foi adoptada uma versão alternativa em que o conjunto de pontos do sinal de entrada é simétrico e centrado do ponto de saída, ou seja, a equação 5.1 toma a forma de:

$$y(n) = \frac{1}{K} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2}} x(n+i) \quad (5.2)$$

O primeiro filtro de média móvel desenhado, tem como objectivo a redução de variações abruptas de cor, sobre regiões suficientemente grandes de modo a permitir uma aproximação à cor de fundo do queijo, e a despistar a presença de olhos no queijo. Este filtro tem uma resposta impulsiva dada pela equação 5.3, com $1/6$ do tamanho da imagem e valores diferentes de zero, onde a amplitude de todos os seus impulsos é igual e com soma unitária, ou seja, se W for o número de pixels da largura da imagem, o número de impulsos da resposta impulsiva é $K = W/6$, pelo que a amplitude de cada impulso é igual a $1/K + 1$. Assume-se que K é um valor par, caso contrário será considerado o próximo valor par, por excesso, relativamente a K . Este filtro será designado por filtro $H_{1/6}$, sendo definido pela equação 5.4.

$$a(n) = \begin{cases} \frac{1}{K+1}, & \text{se } -\frac{K}{2} \leq n \leq +\frac{K}{2} \\ 0, & \text{c.c.} \end{cases} \quad (5.3)$$

$$y_a(n) = a_0 x\left(n + \frac{K}{2}\right) + a_1 x\left(n + \frac{K}{2} - 1\right) + \dots + a_{\frac{K}{2}} x(n) + \dots + a_{-\frac{K}{2}+1} x\left(n - \frac{K}{2} + 1\right) + a_{-\frac{K}{2}} x\left(n - \frac{K}{2}\right) \quad (5.4)$$

onde $a_0, \dots, a_W = 1/K + 1$.

O filtro da equação 5.4, executa a convolução do sinal $x(n)$, valores da componente azul da imagem original, com a resposta impulsiva da equação 5.3. Este filtro produz um sinal mais suave comparativamente ao sinal original, reduzindo as variações bruscas de cor, permitindo uma aproximação do sinal à cor de fundo do queijo. O resultado da aplicação deste filtro, é mostrado na figura 5.4.

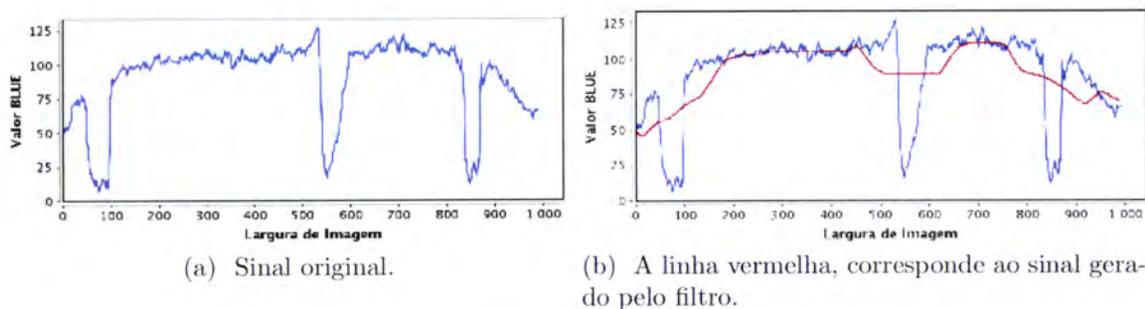


Figura 5.4: Resultado da aplicação do filtro $H_{1/6}$ a uma linha da imagem 5.1a.

Para a detecção dos olhos, será observada a diferença da cor de fundo, aproximada pelo sinal $y_a(n)$, para o sinal original $x(n)$. No entanto, a presença de ruído no sinal $x(n)$, não permite que seja efectuada uma boa segmentação apenas com a subtração de ambos os sinais e posterior comparação com um limiar. Filtrando apenas o sinal $x(n)$ com o filtro $H_{1/6}$, alguns pixels serão classificados erradamente, como pertencentes a regiões escuras, e como tal incluídos nos olhos.

A eliminação ou diminuição da influência do ruído no resultado da segmentação, obrigou ao desenho de um outro filtro. Neste caso, consiste novamente num filtro de média móvel, mas agora com uma resposta impulsiva com $Q = K/10 = W/60$ impulsos. Este filtro designado por filtro $H_{1/60}$ é definido por:

$$b(n) = \begin{cases} \frac{1}{Q+1}, & \text{se } -\frac{Q}{2} \leq n \leq +\frac{Q}{2} \\ 0, & \text{c.c.} \end{cases} \quad (5.5)$$

$$y_b(n) = b_0 x\left(n + \frac{Q}{2}\right) + b_1 x\left(n + \frac{Q}{2} - 1\right) + \dots + b_{\frac{Q}{2}} x(n) + \dots + b_{-\frac{Q}{2}+1} x\left(n - \frac{Q}{2} + 1\right) + b_{-\frac{Q}{2}} x\left(n - \frac{Q}{2}\right) \quad (5.6)$$

Tal como o filtro $H_{1/6}$, no filtro $H_{1/60}$ o número de impulsos da resposta impulsiva é ajustado em função do número de pixels do comprimento da imagem, sendo neste caso $1/60$ do comprimento da mesma. A figura 5.5 apresenta o resultado da aplicação do filtro $H_{1/60}$. Em 5.5c é efectuada um *zoom* de uma secção dos sinais da figura 5.5b, de modo a ser possível observar com mais pormenor o sinal $y_b(n)$, resultante da aplicação do filtro $H_{1/60}$ ao sinal $x(n)$.

O sinal resultante da aplicação do filtro $H_{1/60}$, $y_b(n)$, é um sinal rápido que se ajusta bastante bem ao sinal $x(n)$, o que dificulta a sua visualização na figura 5.5b. A ligeira diferença existente entre o sinal original $x(n)$ e o sinal $y_b(n)$, embora quase imperceptível, permite a eliminação do ruído presente no sinal $x(n)$.

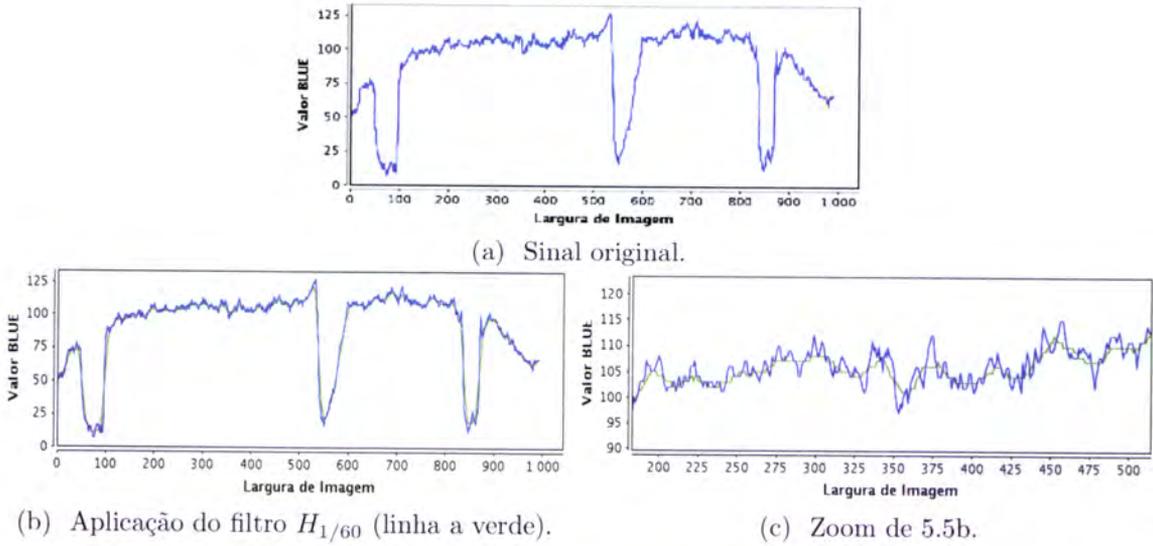


Figura 5.5: Resultado da aplicação do filtro linear.

A conjugação dos dois filtros acima definidos, $H_{1/6}$ e $H_{1/60}$, permite não só a eliminação de ruído como a identificação de oscilações significativas da cor azul nas imagens de queijos, correspondendo estas oscilações às zonas onde se encontram os olhos do queijo. Este processo de filtragem é aplicado em todas as linhas da imagem, seguindo-se a subtração do sinal resultante do filtro $H_{1/60}$ ao sinal resultante do filtro $H_{1/6}$.

$$y_{ab}(n) = y_a(n) - y_b(n) \quad (5.7)$$

Segue-se a comparação do sinal $y_{ab}(n)$ com um limiar (do inglês *threshold*) definido por 5.8, que varia em função da cor de fundo da linha a ser processada, de modo a adaptar o limiar a imagens muito escuras e com pouco contraste entre o fundo do queijo e os seus olhos.

$$th_i = \begin{cases} 10 & se \quad max_i < 15 \\ (2 * max_i) - 20 & se \quad 15 \leq max_i < 20 \\ 20 & se \quad max_i \geq 20 \end{cases} \quad (5.8)$$

onde $max_i = \max_j (I_1(i, j) - I_0(i, j))$, I_0 a imagem original e I_1 uma imagem intermédia resultante dos filtros de média móvel $H_{1/6}$ e $H_{1/60}$.

Dessa comparação obtém-se uma nova imagem, $G(i, j)$, resultante da aplicação da segmentação definida na equação 5.9, segundo a qual se $y_{ab}(n)$ for superior ou igual ao limiar th_i , o pixel da nova imagem toma o valor *null* o que significa que pertence a zonas definidas como olhos do queijo, caso contrário o pixel receberá o valor da componente azul do pixel correspondente na imagem original. Na equação 5.9, $I(i, j)$ representa a imagem original.

$$Th(I(i, j), y_a(n), y_b(n)) = \begin{cases} null & \text{se } y_a(n) - y_b(n) \geq th_i \\ I(i, j) & \text{c.c.} \end{cases} \quad (5.9)$$

Contudo, a tentativa de definição da cor de fundo do queijo pela aplicação do filtro $y_{ab}(n)$, é influenciada pela presença de zonas mais escuras, correspondentes aos olhos. Por isso, de modo a diminuir a influência dos olhos no nível de cor de fundo do queijo, o filtro $y_{ab}(n)$ irá ser aplicado à imagem $G(i, j)$, derivada da segmentação anterior. Este processo permite uma aproximação à verdadeira cor de fundo do queijo, de forma iterativa, melhorando o processo de segmentação.

O próximo filtro a aplicar terá de lidar com valores a *null*. Por isso o filtro $H_{1/6}$ é redefinido, para uma forma não linear, com os coeficientes da resposta impulsiva iguais a

$$a_{-\frac{K}{2}}, \dots, a_{\frac{K}{2}} = \frac{1}{K - N} \quad (5.10)$$

onde N é o número de pixels a *null* da janela do filtro em cada momento. No caso de $K = N$ então $y_a = y_a(n - 1)$, e no caso de $N = 0$ a definição do filtro mantém-se inalterada, tal como a equação 5.4.

Após a aplicação de $H_{1/6}$, numa segunda iteração, é realizada nova segmentação mas agora comparando a subtracção do sinal resultante do filtro $H_{1/60}$ com o sinal filtrado pelo filtro $H_{1/6}$ (após a redefinição do filtro), com um novo limiar calculado novamente por 5.8, obtendo-se um novo limiar.

No entanto, esta segunda segmentação apresenta uma particularidade. Além dos sinais obtidos pelos filtros $H_{1/6}$ e $H_{1/60}$, é necessário ter conhecimento do resultado da segmentação anterior, de modo a que um pixel que tenha sido classificado como pertencente a um olho, não seja classificado como pertencente ao fundo, na segmentação seguinte. Embora tal facto seja assumido como natural, surgiram casos ao longo dos testes efectuados, que demonstram o contrário, tema que será abordado na secção 5.3.2. Portanto, a equação 5.9 foi redefinida, assumindo a forma da equação 5.11.

$$Th(I(i, j), y_a(n), y_b(n), G(i, j)) = \begin{cases} null & \text{se } y_a(n) - y_b(n) \geq th_i \vee G(i, j) = 0 \\ I(i, j) & \text{c.c.} \end{cases} \quad (5.11)$$

A figura 5.6a apresenta o sinal resultante da aplicação da primeira e segunda iteração do filtro $H_{1/6}$, na forma não linear.

Uma vez diminuída a influência dos olhos, resultante da aplicação da filtragem não linear, consegue-se uma melhor aproximação à verdadeira cor de fundo do queijo, permitindo uma melhor definição dos contornos dos olhos.

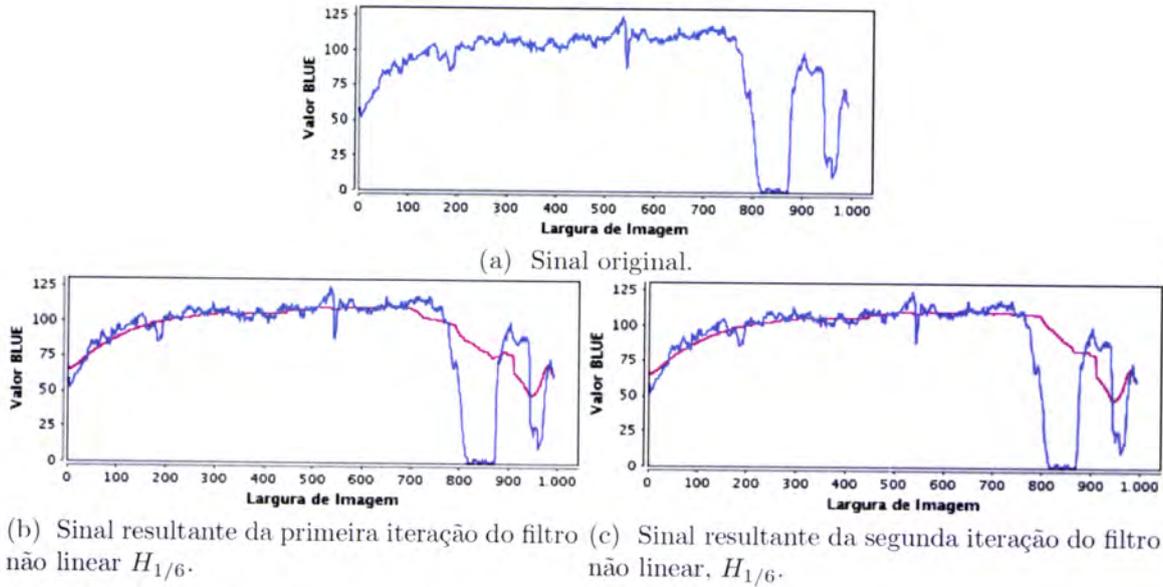


Figura 5.6: Aplicação do filtro $H_{1/6}$ na forma não linear.

5.3 Considerações sobre o algoritmo

5.3.1 Número de iterações da filtragem não linear

De modo a aferir o método de segmentação exposto no ponto anterior, foram utilizadas 34 fotografias de queijos de Évora. A componente não linear dos filtros apresentados, foi desenvolvida tendo em vista a sua utilização de forma iterativa.

Com o aumento do número de iterações do filtro não linear, assumem-se certos pressupostos, de entre os quais se destacam: melhor definição do contorno dos olhos; o acréscimo de área reconhecida como olhos diminui a cada iteração, tendendo para zero; o tempo de processamento aumenta.

Para conseguir um equilíbrio entre o tempo de processamento e o resultado final do algoritmo de segmentação, aplicou-se o algoritmo com diversas iterações às 34 fotografias de teste, tendo-se concluído que para a maioria das fotografias de teste, apenas duas iterações são suficientes para se obter uma segmentação bastante aceitável.

Convencionou-se que a segmentação é considerada aceitável, quando entre duas iterações consecutivas, o acréscimo de área reconhecida como olhos é inferior a 5%. No entanto, na eventualidade de ocorrer algum caso extremo, em que a condição anterior nunca se verifique, foi imposto um limite máximo de 5 iterações.

O gráfico 5.7, resume o comportamento do algoritmo de segmentação relativamente ao acréscimo de área reconhecida como olhos entre duas iterações consecutivas do filtro não linear, verificando-se que em nenhum dos testes foi necessário mais que 3 iterações para que se verificasse um acréscimo inferior a 5%, entre duas iterações consecutivas.

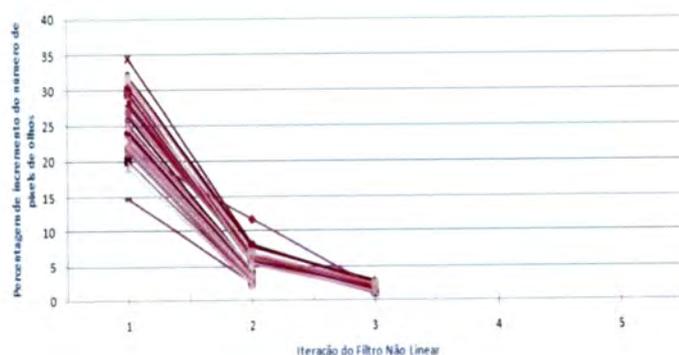
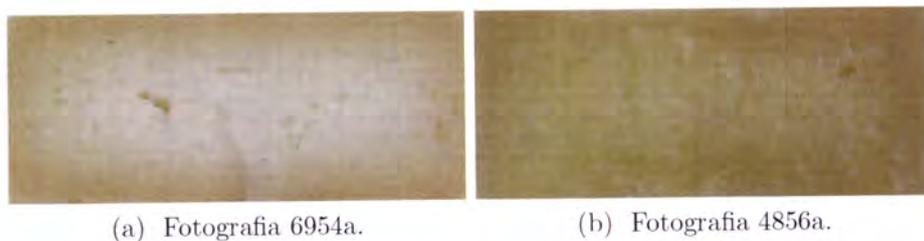


Figura 5.7: Relação entre o número de iterações do filtro não linear e o acréscimo de área reconhecida como olhos do queijo.

5.3.2 Comportamento oscilante

Na secção 5.2, foi referido que a filtragem iterativa de determinadas imagens com filtros não lineares, produzia resultados imprevistos. Admite-se como previsto, que o incremento da área correspondente aos olhos, entre iterações consecutivas do filtro não linear, tende para zero com o aumento do número de iterações. Admite-se igualmente que entre iterações consecutivas do filtro não linear o valor da área dos olhos não diminua.

Se for utilizada a função de segmentação 5.9, pode constatar-se que, para o caso das fotografias da figura 5.8, os pressupostos acima referidos não se verificam.



(a) Fotografia 6954a.

(b) Fotografia 4856a.

Figura 5.8: Imagens em que as iterações dos filtros não lineares provocam um comportamento oscilante.

Esta função produz um resultado oscilatório relativamente à área reconhecida como olhos, o que se verifica pela análise do gráfico da figura 5.9.

Ainda na secção 5.2, foi apresentada uma solução para o problema, passando pela alteração da função de segmentação representada pela equação 5.9 para a forma da equação 5.11, de modo a corrigir o comportamento oscilatório.

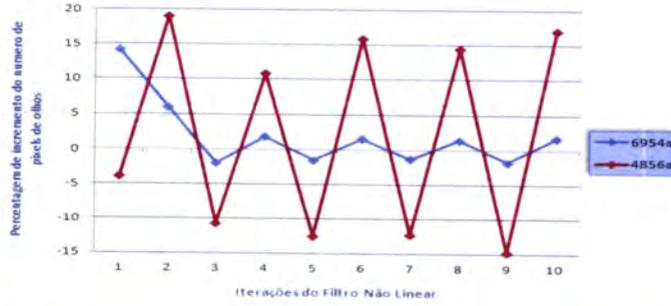


Figura 5.9: Gráfico ilustrativo da oscilação no resultado da segmentação das fotografias 5.8a e 5.8b.

5.3.3 Valores da resposta impulsiva nos limites da imagem

Os filtros descritos na secção 5.2, necessitam de valores exteriores à imagem nos limites laterais da mesma, pelo que, estes valores são calculados segundo as equações 5.12 e 5.13, para o lado esquerdo e direito respectivamente.

$$x_{esquerda}(i, j) = \sum_j^{\frac{k}{2}-j} I(i, j) \times \frac{1}{\frac{k}{2}-j}, \quad j - \frac{k}{2} < i < 0 \quad (5.12)$$

$$x_{direita}(i, j) = \sum_{a=\frac{k}{2}-j}^n I(i, a) \times \frac{1}{\frac{k}{2}-a}, \quad n - (a - \frac{k}{2}) < i < n \quad (5.13)$$

onde n é a largura da imagem em pixels e k o número de impulsos da resposta impulsiva do filtro.

5.4 Resumo do processo de segmentação

O algoritmo de segmentação, poderá ser descrito pelas equações:

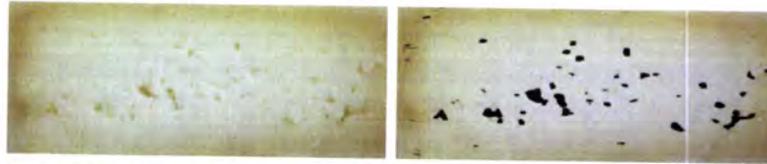
$$G^0(i, j) = Th\left(I(i, j), H_{1/6}(I(i, j)), H_{1/60}(I(i, j)), G^{-1}(i, j)\right) \quad (5.14)$$

$$G^1(i, j) = Th\left(I(i, j), H_{1/6}(G^0(i, j)), H_{1/60}(I(i, j)), G^0(i, j)\right) \quad (5.15)$$

$$G^2(i, j) = Th\left(I(i, j), H_{1/6}(G^1(i, j)), H_{1/60}(I(i, j)), G^1(i, j)\right) \quad (5.16)$$

$$G^m(i, j) = Th\left(I(i, j), H_{1/6}(G^{m-1}(i, j)), H_{1/60}(I(i, j)), G^{m-1}(i, j)\right) \quad (5.17)$$

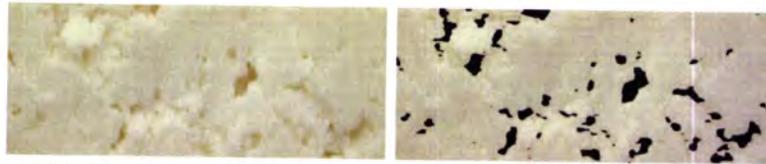
onde $G^{-1}(i, j) = I(i, j)$. De uma forma simplificada, o algoritmo pode ser igualmente descrito por:



(a) Imagem original 8895b (b) Resultado da Segmentação

Figura 5.12: Segmentação da imagem 8895b.jpg.

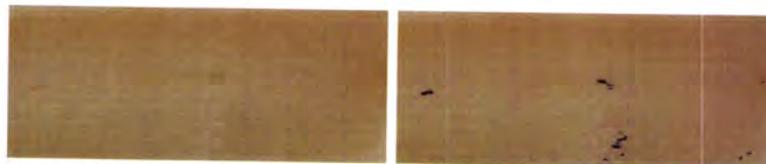
A cor do queijo da figura 5.12 apresenta-se pouco uniforme, sendo mais escuro junto da crosta e mais claro no centro, o que pode indicar que o processo de cura não foi concluído.



(a) Imagem original 7843b (b) Resultado da Segmentação

Figura 5.13: Segmentação da imagem 7843b.jpg.

O queijo da figura 5.13 é claramente um queijo muito branco e por isso muito mal curado, apresentando olhos relativamente grandes, aumentando a percentagem de área coberta por estes. Praticamente todos os critérios de apreciação interna são desrespeitados, pois verifica-se que a pasta não se encontra ligada nem bem fechada, os olhos não são pequenos e a cor não é amarelada e uniforme.



(a) Imagem original 5946a (b) Resultado da Segmentação

Figura 5.14: Segmentação da imagem 5946a.jpg.

Na figura 5.14 o queijo embora apresente uma reduzida percentagem de área correspondente aos olhos e uma cor uniforme, o tom de cor é bastante escuro. Tal facto, pode dever-se a um tempo excessivo do processo de cura.

Os restantes resultados obtidos para todas as fotografias utilizadas no teste ao algoritmo, além de constarem no apêndice A.1, podem igualmente ser consultados em [https://www.alunos.uevora.pt/\(til\)120291/thesis/resultados.html](https://www.alunos.uevora.pt/(til)120291/thesis/resultados.html)

5.6 Comparação com outros algoritmos de segmentação

A avaliação do algoritmo desenvolvido, foi realizada por comparação com outros algoritmos de segmentação referidos na literatura e implementados na aplicação ImageJ. De entre os algoritmos testados destacam-se os algoritmos *Otsu Threshold*¹, *Maximum Entropy Threshold*² e *K-Means Clustering*³.

5.6.1 *Otsu Threshold*

O método desenvolvido por Nobuyuki Otsu [48], é um dos métodos mais eficazes e por isso mais utilizados para a segmentação de imagens recorrendo a técnicas de limiarização (*thresholding*) [22].

O método de Otsu pode ser implementado em 1D, 2D e 3D, no entanto, importa apenas considerar no presente texto a vertente 1D, visto que o algoritmo desenvolvido processa a imagem apenas a uma dimensão. Na versão 1D, o método de Otsu, após analisar o histograma dos níveis de cinza da imagem, selecciona o limiar global óptimo, que maximiza a variância entre as classes.

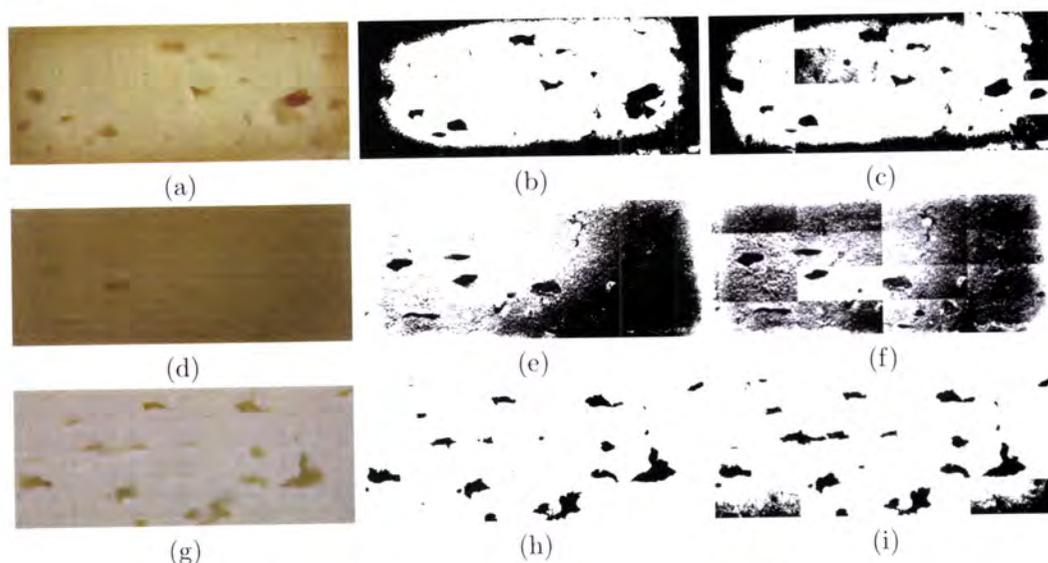


Figura 5.15: Aplicação do algoritmo *Otsu Threshold*.

Tendo em consideração que o algoritmo desenvolvido nesta tese aplica um limiar, eventualmente diferente para cada linha da imagem, foi pensada uma forma de possibilitar ao

¹Os resultados apresentados baseiam-se no código retirado de <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

²Plugin do ImageJ disponível em <http://rsbweb.nih.gov/ij/plugins/entropy.html>

³http://sourceforge.net/projects/ij-plugins/files/ij-plugins_toolkit

método de Otsu uma melhor segmentação, pelo cálculo de 16 limiares ao invés de apenas 1. A imagem inicial é dividida em 16 partes iguais, sendo aplicado o método de Otsu a cada uma das sub-imagens, pelo que serão calculados 16 limiares, sendo posteriormente agregadas numa nova imagem o resultado das diversas segmentações tal como é mostrado nas imagens 5.15c, 5.15f e 5.15i. As imagens 5.15b, 5.15e e 5.15h são o resultado do método de Otsu quando este calcula apenas um limiar para toda a imagem.

Verifica-se que a intenção de proporcionar uma melhor adaptação ao método *Otsu Threshold* não resulta num incremento significativo na qualidade da segmentação.

Na figura 5.16, mostra-se a comparação da segmentação realizada pelo método de Otsu e o algoritmo apresentado nesta tese, para uma linha de cada uma das fotografias 5.15a, 5.15d e 5.15g.

O sinal a azul claro corresponde à variação de cor em escala de cinza, visto que o método de Otsu utiliza apenas a informação de cor na escala de cinza. Os segmentos de linha a verde, correspondem aos quatro limiares que o método de Otsu define para essa linha da imagem, tendo em consideração a divisão da imagem em 16 regiões com as mesmas dimensões.

A azul escuro encontra-se o sinal correspondente à componente azul do espaço de cores RGB e a roxo o sinal resultante da aplicação da filtragem pelo método de segmentação apresentado neste capítulo.

Observa-se que na generalidade dos casos, a segmentação pelo método de Otsu não define de forma rigorosa os contornos dos olhos, estando muito susceptível ao ruído.

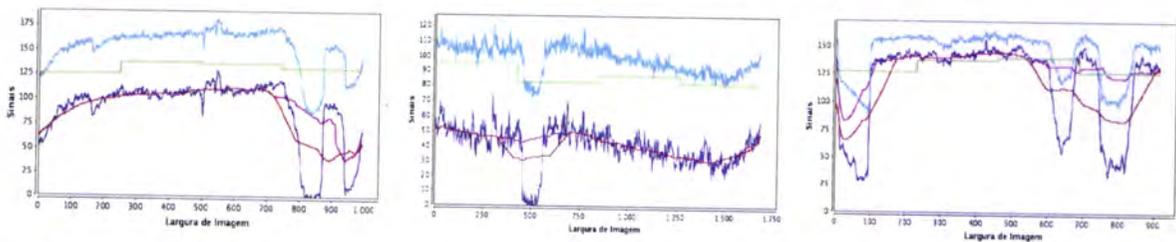


Figura 5.16: Comparação da segmentação desenvolvida com o método de Otsu.

Na tabela A.2, do apêndice A.3, são apresentados os valores dos 16 limiares calculados pelo método de Otsu, correspondentes ao limiar para cada região aquando da divisão da imagem em 16 regiões e o valor do limiar quando uma imagem é considerada na sua totalidade. O limiar total encontra-se relativamente próximo da média dos 16 limiares parciais.

5.6.2 *Maximum Entropy Threshold*

O método *Maximum Entropy Threshold* permite segmentar uma imagem, recorrendo a uma limiarização automática baseada na entropia do histograma. O método é muito semelhante ao método de Otsu 5.6.1. Em vez de maximizar a variância inter-classe, maximiza a entropia inter-classe. Entropia é uma medida da incerteza de um determinado acontecimento ocorrer, dada por:

$$H(X) = - \sum_{i,j} p_{i,j} \times \log_2(p_{i,j}) \quad (5.20)$$

O valor da entropia é bastante simples de calcular, tendo por base os dados do histograma, onde $p_{i,j}$ representa a probabilidade do valor de um pixel na escala de cinzentos.

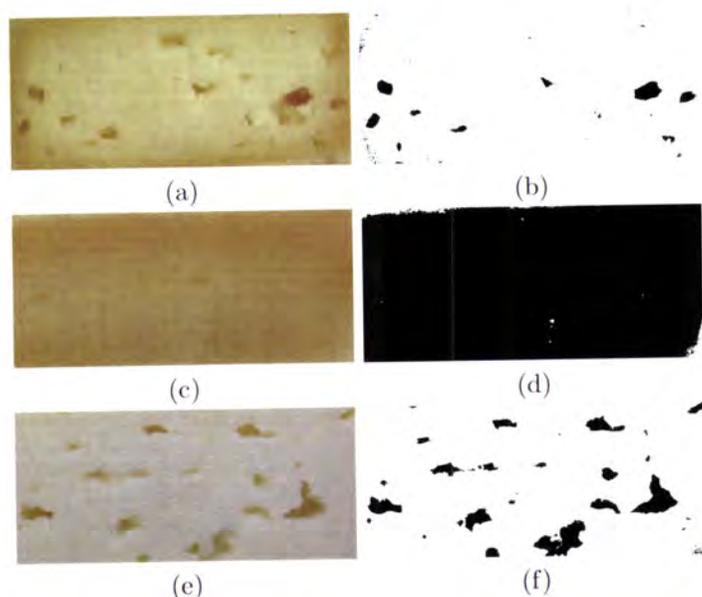


Figura 5.17: 5.17b, 5.17d, 5.17f segmentação com o algoritmo *Maximum Entropy Threshold*, das imagens 5.17a, 5.17c, 5.17e, respectivamente.

5.6.3 *K-Means Clustering*

O algoritmo *K-Means Clustering* é amplamente utilizado em diversas áreas, tais como, inteligência artificial, biologia, mineração de dados, compressão de dados, processamento de imagem, medicina [61], entre outros. É um algoritmo de aprendizagem não-supervisionada, que consiste na divisão do conjunto de dados em K agrupamentos (*clusters*), por um processo iterativo.

O processo da definição dos *clusters* compreende duas etapas: inicialmente são calculados K centros aleatórios, onde K é um valor pré-definido; a próxima etapa aproxima cada dado do centro mais próximo, sendo a distância Euclideana a medida geralmente utilizada para o cálculo da distância entre os dados e o centro do *cluster*. A segunda etapa é repetida iterativamente, aproximando os dados de cada *cluster* ao seu centro, até que seja atingido um critério pré-estabelecido de distância mínima.

Outras distâncias podem ser consideradas para o cálculo referido no parágrafo anterior, como por exemplo, a distância de Mahalanobis [8], no entanto, no âmbito do processamento

de imagem, a distância entre as coordenadas dos pixels, a cor RGB ou a textura da imagem, são as medidas mais utilizadas.

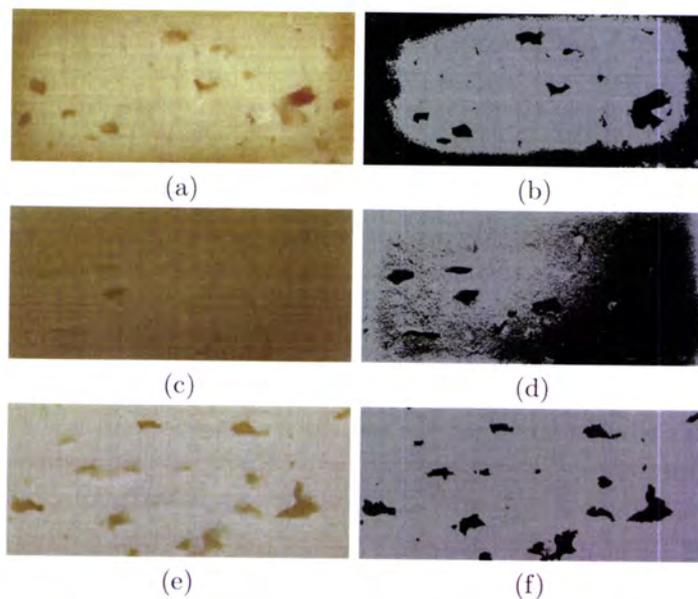


Figura 5.18: 5.18b, 5.18d, 5.18f segmentação com o algoritmo *K-Means Clustering*, das imagens 5.18a, 5.18c, 5.18e, respectivamente.

Capítulo 6

Detecção de Texturas

"... if food quality is to be assessed, it should be remembered that the internal food structure may have to be deduced in some way from the appearance, and the method of achieving this will not be known a priori."

E. R. Davies [20]

6.1 Introdução

Neste capítulo é apresentada a metodologia desenvolvida para a detecção de textura na crosta exterior do queijo. A textura apresenta a forma de uma malha de quadrados, devido à forma dos cestos onde os queijos são colocados a secar. A ocorrência de textura pode indicar que o processo de secagem do queijo ocorreu demasiado depressa ou que o queijo foi pouco compactado.

Este capítulo é baseado no relatório da disciplina de Sistemas de Decisão e Controlo por Computador, no âmbito da qual foram desenvolvidos os algoritmos apresentados ao longo deste capítulo.

6.2 Textura

O conceito de textura não apresenta uma definição clara e objectiva, tornando-se extremamente difícil encontrar exemplos de boa ou má textura, visto que a textura é utilizada maioritariamente como indicador de outro parâmetro mais significativo [40].

Numa definição de dicionário, a textura refere-se à disposição dos componentes de um material, assumindo maior importância a natureza do material, ao invés do seu gosto, da sua aparência ou do seu "sentir" [20].

Em [19], a textura é caracterizada pela variação de intensidade da cor que normalmente origina a rugosidade da superfície do objecto. A obtenção de uma textura bem definida, pode

apresentar variações de intensidade tanto regulares como aleatórias, e por isso, a análise de texturas tem-se baseado em informações/medidas estatísticas.

Têm sido desenvolvidas diversas metodologias para a análise e detecção de texturas, nomeadamente a abordagem Laws' [38] (1980) baseada na matriz de co-ocorrência. Na sequência da abordagem de Laws', Ade's em 1983 desenvolveu outra metodologia [4], que com a utilização de filtros de máscara, secção 3.5, permite uma análise mais cuidada das componentes da textura. Outras abordagens são apresentadas em [77] e [44]. No entanto, a metodologia mais utilizada é a descrita em [54].

6.3 Transformada de Hough

A transformada de Hough (TH), desenvolvida por Paul Hough em 1960 [27], é um algoritmo utilizado para efectuar a detecção de figuras geométricas que possam ser descritas por simples equações com apenas alguns parâmetros, como por exemplo, linhas [46], [64], círculos [36], elipses [15] ou formas compostas dessas estruturas básicas.

Têm sido diversos os domínios onde a transformada de Hough é aplicada, como por exemplo, no reconhecimento da iris do olho humano [65] ou detecção de bolachas [20].

A transformada de Hough é usada no âmbito deste trabalho para detecção de linhas, com o intuito de detectar a presença da malha de quadrados que caracteriza a textura da crosta exterior do queijo de Évora.

6.3.1 Transformada de Hough para detecção de rectas

A transformada de Hough é talvez o algoritmo mais utilizado para a detecção e ligação de segmentos de linha [71], envolvendo a modificação da representação de uma linha no espaço de coordenadas cartesianas para o espaço de coordenadas polares. Uma linha no plano 2D pode ser descrita com dois parâmetros, usando a equação reduzida de uma recta:

$$y = kx + d \quad (6.1)$$

em que k representa o declive da recta e d o ponto de intersecção da recta no eixo das ordenadas.

Uma recta que passa por dois pontos genéricos $p_1 = (x_1, y_1)$ e $p_2 = (x_2, y_2)$, deve satisfazer as condições da equação 6.2, representadas na figura 6.1¹.

$$\begin{aligned} y_1 &= kx_1 + d \\ y_2 &= kx_2 + d \quad , k, d \in \mathfrak{R} \end{aligned} \quad (6.2)$$

¹Adaptado de [71].

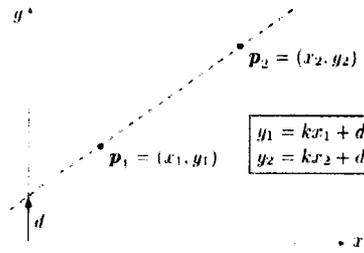


Figura 6.1: Definição gráfica de pontos colineares.

Considerando as equações 6.2, o objectivo será encontrar valores para os parâmetros k e d tal que sobre esta recta se encontrem grande parte dos pontos que pertencem ao contorno do objecto que se pretende identificar na imagem [71]. Uma abordagem possível, dado a natureza discreta de uma imagem, é gerar exaustivamente todos os segmentos de recta possíveis e contar o número de pixels que estão sobrepostos à recta definida pelos pontos p_1 e p_2 . Facilmente se compreende que é um processo bastante ineficiente e muito exigente em termos computacionais.

A transformada de Hough, segue noutra direcção, examinando todas as rectas que podem passar por um determinado ponto da imagem, pelo que todas as linhas L_j que possam passar através de um ponto $p_0 = (x_0, y_0)$, figura 6.2², devem satisfazer a condição 6.3.

$$L_j : y_0 = k_j x_0 + d_j \quad \forall k_j, d_j \quad (6.3)$$

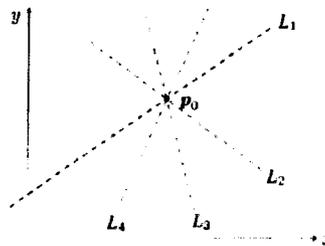


Figura 6.2: Conjunto de rectas que passam num ponto.

Tal como na abordagem anterior, existe um conjunto infinito de soluções para k_j e d_j , o que corresponde a um conjunto infinito de linhas que podem passar pelo ponto p_0 . Dado k_j , obtém-se d_j por:

$$d_j = -x_0 k_j + y_0 \quad (6.4)$$

configurando-se como outra representação da equação de uma recta, sendo neste caso k_j e d_j as variáveis e (x_0, y_0) os parâmetros constantes da equação. As soluções para k_j e d_j descrevem todas as linhas que passam no ponto $p_0 = (x_0, y_0)$.

²Adaptado de [71].

Considerando um ponto arbitrário da imagem $p_i = (x_i, y_i)$, a equação 6.4 descreve a linha $M_i : d = -x_i k + y_i$ sendo os parâmetros $(-x_i, y_i)$ designados por parâmetros do espaço de Hough.

Tabela 6.1: Relação entre o espaço da imagem e o espaço de parâmetros de Hough.

Espaço da Imagem (x,y)		Espaço de Parâmetros (k,d)	
Ponto	$p_i = (x_i, y_i)$	Linha	$M_i : d = -x_i k + y_i$
Linha	$L_j : y = k_j x + d_j$	Ponto	$q_j = (k_j, d_j)$

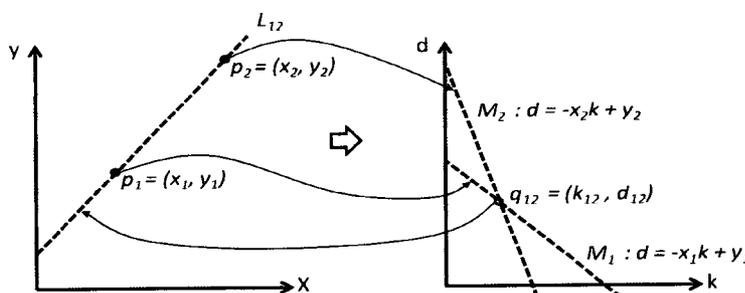


Figura 6.3: Relação entre o espaço da imagem e o espaço de parâmetros.

A figura 6.3³ representa a relação entre o espaço da imagem e o espaço de parâmetros, em que cada ponto $q_{ij} = (k_{ij}, d_{ij})$ no espaço de parâmetros corresponde a uma linha L_{ij} no espaço da imagem. A intersecção das duas linhas M_1 e M_2 no ponto $q_{12} = (k_{12}, d_{12})$ do espaço de parâmetros indica que a linha L_{12} passa pelos pontos k_{12} e d_{12} que existem no espaço da imagem.

Então, se N linhas intersectam o ponto $q' = (k', d')$ no espaço de parâmetros, logo existem N pontos na imagem que estão sobre a linha $y = k'x + d'$ no espaço da imagem.

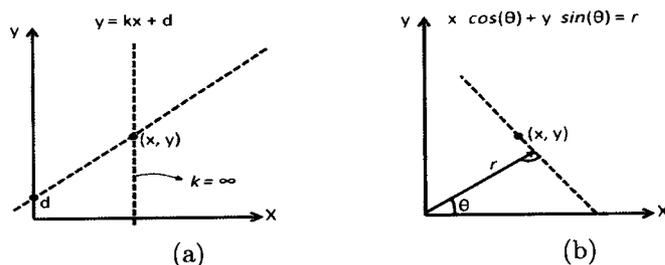


Figura 6.4: Representação de linhas em 2D.

³Adaptado de [71].

A equação 6.1 não poderá ser utilizada como definição de uma recta no âmbito da aplicação da transformada de Hough, visto que, para linhas verticais o declive será $k = \infty$. A solução passa por representar as rectas na Forma Normal de Hessian⁴ (*Hessian Normal Form*) [29],

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = r \quad (6.5)$$

onde r é o comprimento da recta e θ o ângulo entre o eixo das abcissas e a recta que passa pela origem, figura 6.4b⁵. Com esta representação, o espaço de parâmetros passa a ser definido pelas coordenadas θ e r , sendo a transformação de um ponto $p_i = (x_i, y_i)$ no espaço imagem para o espaço de parâmetros, dado por

$$r_{x_i, y_i}(\theta) = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad 0 \leq \theta \leq \pi \quad (6.6)$$

Se se utilizar o centro da imagem como ponto de referência para o espaço da imagem, pode-se limitar o intervalo de valores de θ , a metade da diagonal da imagem

$$-r_{max} \leq r_{x,y}(\theta) \leq r_{max}, \quad \text{onde } r_{max} = \frac{\sqrt{W^2 + H^2}}{2} \quad (6.7)$$

com largura W e altura H .

6.3.2 Transformada de Hough para outras figuras geométricas

A transformada de Hough para detecção de rectas, descrita na secção anterior, pode ser extendida para o reconhecimento de outras figuras geométricas. Para tal, é necessário determinar a equação paramétrica da figura que se deseja detectar, tendo em consideração que a quantidade de parâmetros está directamente relacionada com a dimensão do espaço de Hough.

Para a representação de um círculo em 2D são necessários três parâmetros, *Circulo* = (\bar{x}, \bar{y}, r) , onde \bar{x} e \bar{y} são as coordenadas do centro e r o raio do círculo. O cálculo dos valores (\bar{x}, \bar{y}) para um determinado ponto $p = (x, y)$, pode ser efectuado a partir da equação 6.8.

$$r^2 = (x - \bar{x})^2 + (y - \bar{y})^2 \quad (6.8)$$

No entanto, esta equação não permite uma implementação eficiente, [26], sendo preferível uma representação em coordenadas polares:

$$\begin{aligned} \bar{x} &= x - r * \cos(\theta) \\ \bar{y} &= y - r * \sin(\theta) \end{aligned} \quad (6.9)$$

⁴A representação de uma recta pela Forma Normal de Hessian, pode ser obtida pela multiplicação da expressão $A_x + B_y + C = 0$, equação geral de uma recta, pelo factor de normalização $\mu = \pm \frac{1}{\sqrt{A^2 + B^2}}$ [29].

⁵Adaptado de [71].

Portanto, a transformada de Hough para círculos requer um espaço de parâmetros a três dimensões de modo a encontrar a posição e o raio dos círculos presentes numa imagem.

No caso de elipses, são necessários cinco parâmetros, $Ellipse = (\bar{x}, \bar{y}, r_a, r_b, \alpha)$ onde (\bar{x}, \bar{y}) são a coordenadas do ponto central, (r_a, r_b) são os dois raios e α é a orientação do eixo principal. A equação paramétrica de uma elipse é dada por

$$\begin{aligned} x(a) &= \bar{x} + \cos(\theta) \cdot r_a \cdot \cos(a) - \sin(\theta) \cdot r_b \cdot \sin(a) \\ y(a) &= \bar{y} + \sin(\theta) \cdot r_a \cdot \cos(a) + \cos(\theta) \cdot r_b \cdot \sin(a) \end{aligned} \quad (6.10)$$

com $0 \leq a \leq 2\pi$.

Para encontrar elipses de qualquer tamanho, posição e orientação são então necessários 5 parâmetros para o espaço de Hough, tornando este método bastante ineficiente. Para estes casos, outra alternativa é utilizar a transformada de Hough generalizada, que permite detectar qualquer figura em 2D [30].

6.3.3 Algoritmo da transformada de Hough

O pseudo-código 3, descreve o algoritmo da transformada de Hough.

Algorithm 3 Transformada de Hough (Adaptado de [71])

```

1: procedure HOUGHLINES( $I(i, j)$ )
2:    $Acc[\theta, r] \leftarrow 0$  ▷ Inicializar a zero o array bidimensional  $Acc$ 
3:    $(i_c, j_c) \leftarrow Centro(I(i, j))$  ▷ Coordenadas do centro da imagem  $I(i, j)$ 
4:   for all As coordenadas  $(i, j)$  do
5:     if  $I(i, j)$  é um ponto de contorno then
6:        $(x, y) \leftarrow (i - i_c, j - j_c)$ 
7:       for all  $\theta_k = 0 \dots \pi$  do
8:          $r_k = x \cdot \cos(\theta_k) + y \cdot \sin(\theta_k)$ 
9:         Incrementar  $Acc[\theta_k, r_k]$ 
10:      end for
11:    end if
12:  end for
13:  Retornar as linhas com os pares de parâmetros  $[\theta_k, r_k]$  para K linhas:
14:   $MaxLinhas \leftarrow EncontrarMaxLinhas(Acc, K)$ 
15:  return MaxLinhas
16: end procedure

```

6.4 Metodologia adoptada para detecção de textura

Segundo [47], o queijo de Évora deve apresentar no seu exterior uma textura lisa ou ligeiramente rugosa. Nesta textura são bem visíveis linhas rectas, que, na grande maioria dos casos formam quadrados, como é apresentado na figura 6.5.



Figura 6.5: Exemplos de queijos que apresentam textura na sua crosta.

De modo a identificar este tipo de textura, foi aplicada a transformada de Hough a imagens como as que são apresentadas na figura 6.5, com o objectivo de detectar a presença das linhas que formam os quadrados e que por sua vez caracterizam a textura.

Tal como foi referido na secção 4.3, do capítulo 4, o processo de detecção de objectos em imagens não é simples de realizar de forma automática [52], pelo que antes de aplicar a transformada de Hough, deve-se proceder à detecção dos contornos dos objectos, utilizando um dos seguintes algoritmos, operadores de Roberts [55], operadores de Sobel [45], operadores de Canny [12, 13] ou operadores de Prewitt [69].

Os operadores de Sobel, também designados por filtro de Sobel, efectuem o cálculo do gradiente (ver equação 4.4) da luminosidade de cada pixel da imagem, informando sobre como varia a luminosidade ao longo da imagem. As variações bruscas de cor, normalmente correspondem aos limites de objectos, pelo que, os contornos dos objectos são identificados por uma nítida alteração de cor ao longo de determinada orientação.

Em termos matemáticos, o cálculo dos operadores de Sobel recorre a filtros lineares, que utilizam duas matrizes 3×3 com o intuito de neutralizar a sensibilidade ao ruído por parte do operador de gradiente simples. As duas matrizes, da equação 6.11, são convolvidas com a imagem original, $I(u, v)$ por forma a calcular as variações de cor, ao longo da direcção horizontal e vertical.

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{e} \quad H_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (6.11)$$

A figura 6.6 mostra o resultado da aplicação dos operadores de Sobel às imagens da figura 6.5.

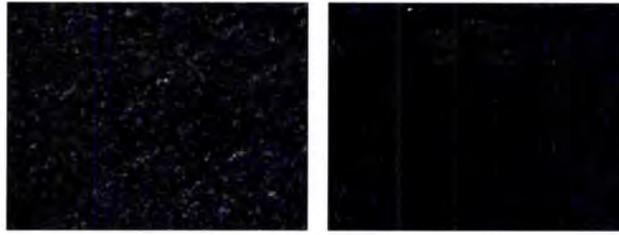


Figura 6.6: Aplicação dos operadores de Sobel.

Sobre a imagem resultante da aplicação dos operadores de Sobel, é então aplicada a transformada de Hough. No algoritmo implementado em MATLAB, apenas são determinadas 10 rectas, pois considerou-se ser um número razoável e que permitia alcançar o objectivo desejado. O resultado da transformada de Hough sobre as imagens da figura 6.6, são mostrados na figura 6.7.

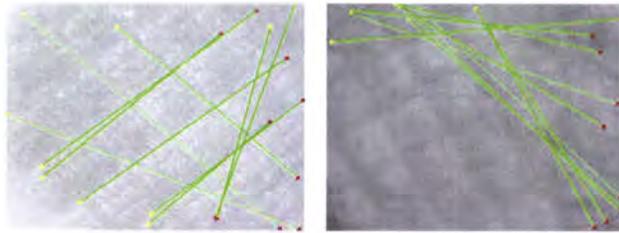


Figura 6.7: Aplicação da transformada de Hough.

6.4.1 Função de avaliação

Tal como foi referido anteriormente, a presença de uma textura bastante rugosa na crosta do queijo de Évora, é um aspecto negativo na sua avaliação. Portanto, visto que a presença de textura prejudica a classificação de um queijo, ao contrário da ausência de textura que beneficia a qualidade do queijo, foi necessário desenvolver um mecanismo que dado as 10 linhas encontradas pela transformada de Hough, não só determinasse a existência de textura, como penalizasse o queijo com textura.

Uma vez que a textura encontrada em imagens do exterior de queijos se assemelha a quadrados, pretende-se verificar se de entre as 10 linhas retornadas pela transformada de Hough, existem linhas paralelas e linhas perpendiculares entre si. O algoritmo da transformada de Hough, apresentado na secção 6.3.3, retorna o ângulo de inclinação de cada recta, pelo que, se a diferença entre os ângulos de inclinação de duas rectas for 0 graus ou múltiplo de 90 graus, tem-se o caso de rectas paralelas ou perpendiculares.

Recorrendo à função *coseno*, figura 6.8, é simples definir uma fórmula de avaliação, que atribui uma pontuação directamente proporcional ao número de rectas paralelas ou perpendiculares.

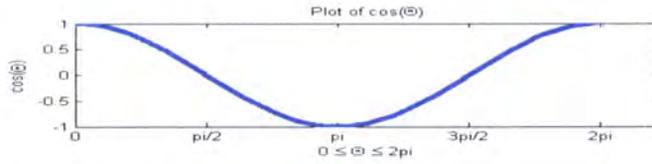


Figura 6.8: Gráfico da função *coseno*.

A fórmula de avaliação, definida na equação 6.12 e ilustrada na figura 6.9, atribui pontuação 1 se o ângulo entre duas rectas for 0 ou múltiplo de 90 graus, baixando progressivamente a pontuação até zero, para os restantes valores dos ângulos.

$$\text{Avaliação}(\theta) = \left(\frac{1 + \cos\left(\frac{8\pi}{360} * \theta\right)}{2} \right)^4 \tag{6.12}$$

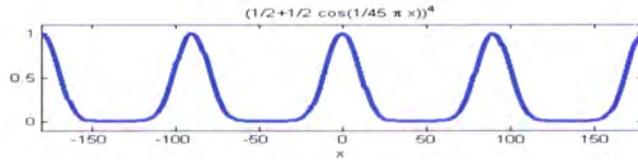


Figura 6.9: Gráfico da função de avaliação.

6.4.2 Escolha do limiar

Tendo em consideração que os melhores queijos são os que não apresentam textura, então quanto menor for o valor de avaliação dado pela equação 6.12, melhor será o queijo.

Observando os resultados apresentados na tabela B.1, verifica-se que as figuras B.10, B.11, B.12, B.15 são os melhores queijos. Portanto, o valor máximo de avaliação que inclui estes quatro queijos é 17, tendo por base os valores registados na tabela B.1.

No entanto, o limiar de avaliação igual a 17, origina a que os queijos das imagens B.7, B.9, B.14 sejam considerados como não apresentando textura, o que não é de todo verdade.

6.4.3 Algoritmo implementado

O algoritmo descrito pelo pseudo-código 7, efectua o cálculo da pontuação de um queijo presente na imagem $I(i, j)$, segundo a equação 6.12.

Algorithm 4 Detecção de Textura

```

1: procedure TEXTURA( $I(i, j)$ )
2:    $G(i, j) \leftarrow EscalaCinza(I(i, j))$ 
3:    $BW \leftarrow ContornosSobel(G(i, j))$ 
4:    $[H, THETA, RHO] \leftarrow Hough(BW)$ 
5:    $Picos \leftarrow Houghpeaks(H)$ 
6:    $Linhas \leftarrow HoughLines(BW, THETA, RHO, Picos)$ 
7:    $Thetas \leftarrow Linhas.THETA$ 
8:    $l \leftarrow 0$ 
9:   for  $k = 1, k = tamanho(Thetas) - 1$  do
10:    for  $l = k + 1, k = Tamanho(Thetas)$  do
11:       $diff[j] \leftarrow Thetas(k) - Thetas(l)$ 
12:       $l \leftarrow l + 1$ 
13:    end for
14:  end for
15:   $Resultado \leftarrow 0$ 
16:  for  $k = 1, k = Tamanho(diff)$  do
17:     $m \leftarrow ((1 + \cos((8 \times \pi/360) \times diff(k)))/2)^4$ 
18:     $Resultado \leftarrow Resultado + m$ 
19:  end for
20:  return Resultado
21: end procedure

```

6.5 Resultados

Tendo em conta o limiar de avaliação igual a 17, nem todos os queijos são correctamente classificados tendo em consideração a presença ou não de textura. Os queijos das figuras B.10, B.11, B.12, B.15 por terem um valor de avaliação inferior a 17 e porque visualmente não apresentam qualquer textura, são de facto os melhores queijos de entre o conjunto de teste.

Os queijos das imagens B.7, B.9, B.14 são incorrectamente classificados como não tendo textura.

6.6 Abordagem inicial

A abordagem inicial ao problema em análise, foi tentar realizar a detecção da textura pelo reconhecimento de padrões em sinais digitais, usando técnicas de processamento de sinal. Estes sinais correspondem ao valor das componentes de cor RGB, dos pixels sobre circunferências centradas no centro do queijo e com raio R .

Para tal, é necessário o cálculo do centro de massa da imagem, que corresponde aproximadamente ao centro do queijo. Após determinar o centro do queijo, são calculadas diversas circunferências centradas no centro do queijo, diferindo no valor do raio.

6.6.1 Cálculo do centro de massa

O centro de massa, (\bar{x}, \bar{y}) , é definido por:

$$\begin{aligned}\bar{x} &= \frac{\sum_{x=1}^n \sum_{y=1}^n x \cdot a(x,y)}{\sum_{x=1}^n \sum_{y=1}^n a(x,y)} \\ \bar{y} &= \frac{\sum_{x=1}^n \sum_{y=1}^n y \cdot a(x,y)}{\sum_{x=1}^n \sum_{y=1}^n a(x,y)}\end{aligned}\quad (6.13)$$

onde $a(x, y)$ corresponde ao valor na escala de cinzentos, do pixel na posição (x, y) .

A figura 6.10b apresenta o ponto calculado como sendo o centro de massa da fotografia 6.10a, aproximadamente o centro do queijo.

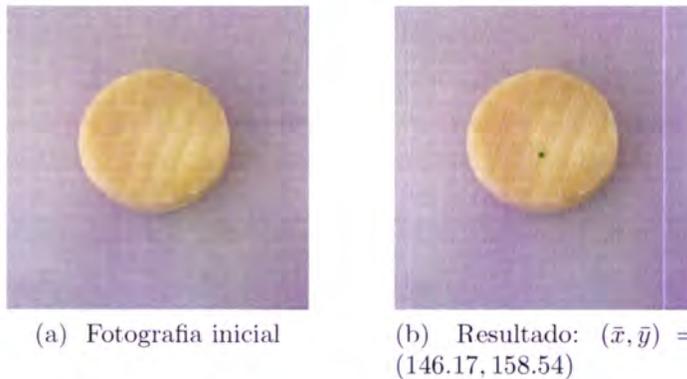


Figura 6.10: Resultado da aplicação do algoritmo para o cálculo do centro de massa.

No pseudo-código 5, é identificado o método de cálculo do centro de massa de uma imagem, simulado em MATLAB.

Algorithm 5 Cálculo do centro de massa

```

1: procedure CENTROMASSA( $I(i, j)$ )
2:    $[x, y, z] \leftarrow \text{Tamanho}(I(i, j))$ 
3:   if  $z \neq 1$  then
4:      $G(i, j) \leftarrow \text{EscalaCinza}(I(i, j))$ 
5:   end if
6:    $[\text{linhas}, \text{colunas}] \leftarrow \text{Tamanho}(G(i, j))$ 
7:    $x = \text{Matriz em que cada conjunto de pixels tem a sua coordenada } x$ 
8:    $y = \text{Matriz em que cada conjunto de pixels tem a sua coordenada } y$ 
9:    $\text{area} = \text{soma}(\text{soma}(G(i, j)))$ 
10:   $\tilde{x} = \text{soma}(\text{soma}(G(i, j)) \cdot x) / \text{area}$ 
11:   $\tilde{y} = \text{soma}(\text{soma}(G(i, j)) \cdot y) / \text{area}$ 
12:  return  $(\tilde{x}, \tilde{y})$ 
13: end procedure

```

6.6.2 Circunferência centrada no centro de massa

Após o cálculo aproximado do centro do queijo, (\tilde{x}, \tilde{y}) , são estimadas circunferências centradas em (\tilde{x}, \tilde{y}) e raio R . As circunferências são determinadas segundo as equações 6.14 e 6.15:

$$x = \tilde{x} + R \times \text{sen}(\alpha) \quad (6.14)$$

$$y = \tilde{y} + R \times \text{cos}(\alpha) \quad (6.15)$$

onde $0 \leq \alpha \leq 2 \times \pi$.

O procedimento de estimação dos valores dos pixels das circunferências, é descrito no pseudo-código 6.

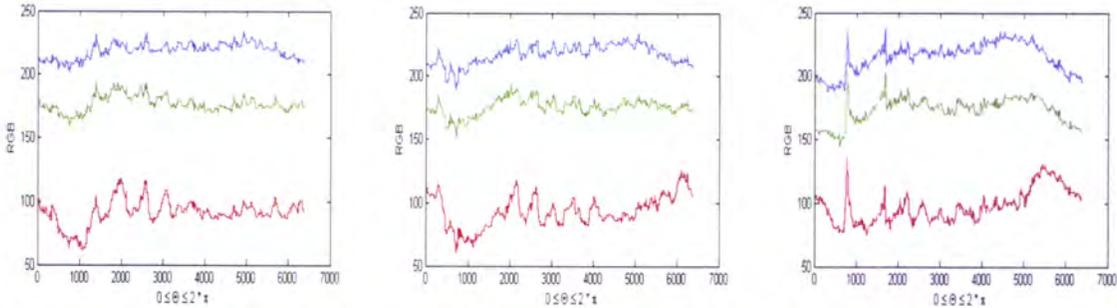
Algorithm 6 Valores RGB da circunferência de raio R , centrada no centro de massa

```

1: procedure CIRCUNFERENCIA( $\tilde{x}, \tilde{y}, R$ )
2:    $[thetas] \leftarrow [0 : 0.001 : 2 * \pi]$ 
3:   for  $i = 1, i = Tamanho(thetas)$  do
4:      $x(i) = \tilde{x} + R * \sin(thetas(i))$ 
5:      $y(i) = \tilde{y} + R * \cos(thetas(i))$ 
6:   end for
7:    $pixel = Pixel(RGB, x, y)$   $\triangleright Pixel(I, x, y)$  função que retorna um tuplo  $[r, g, b]$ 
   correspondente às coordenadas RGB do ponto  $(x, y)$ 
8:    $Grafico(pixel)$ 
9: end procedure

```

Visualizando os sinais correspondentes à variação da intensidade de cor dos pixels das circunferências, para valores de raio diferentes, figura 6.11, observa-se que os sinais não apresentam a repetição de um padrão que possa indicar a presença de textura.



(a) Circunferência com raio 45. (b) Circunferência com raio 50. (c) Circunferência com raio 55.

Figura 6.11: Sinal RGB dos valores amostrados para circunferências com raios diferentes, centradas no centro de massa.

Visto que, não se vislumbrou a possibilidade de detectar a presença de textura pela metodologia acima referida, esta foi posta de parte, optando-se pela utilização da transformada de Hough, que tal como foi exposto na secção 6.4, permite detectar a presença de textura em imagens da crosta externa de queijos.

Capítulo 7

Recolha de Atributos

"Extraction is more complex than detection, since extraction implies that we have a description of a shape, such as its position and size, whereas detection of a shape merely implies knowledge of its existence within an image."

M. S. Nixon, A. S. Aguado [41]

7.1 Introdução

No presente capítulo serão definidos os atributos das imagens dos queijos usadas na classificação e avaliação. Serão igualmente descritos os algoritmos usados na extracção e cálculo dos diversos atributos.

7.2 Atributos relativos aos olhos

Na sequência do algoritmo de segmentação apresentado no capítulo 5, são obtidas imagens binárias, em que o branco é representado o que é considerado fundo do queijo e o preto os olhos do queijo, tal como em 7.1a, 7.1c.

Estas imagens são posteriormente processadas pelo algoritmo *Analyze Particles*¹ disponível na aplicação ImageJ, que tal como o nome indica, analisa e extrai características de partículas reconhecidas numa imagem binária, partículas essas que consistem num conjunto de pixels conexos.

Visto que as partículas presentes nas imagens a processar pelo algoritmo acima referido, são os olhos do queijo, foram seleccionadas apenas as características e informações relevantes tendo em consideração a análise que se pretendia fazer das imagens, nomeadamente, o número de olhos, o seu tamanho médio, a fracção da área ocupada pelos olhos no queijo

¹Disponível em <http://rsbweb.nih.gov/ij/plugins/index.html>

e a área total em pixels, que corresponde exactamente ao valor calculado aquando da segmentação.

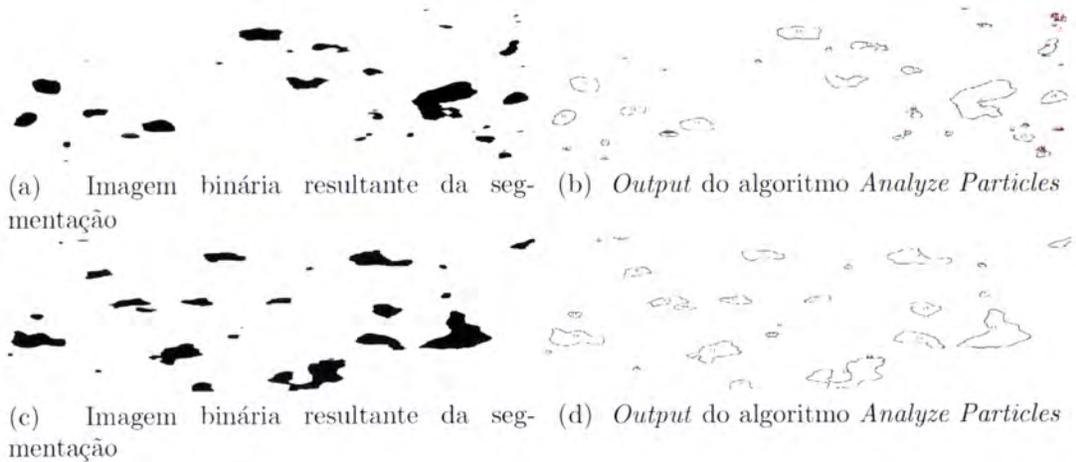


Figura 7.1: Aplicação do *plugin Analyze Particles*, do ImageJ.

Além das características acima referidas, foi tido também em consideração o perímetro de cada olho, de modo a possibilitar o cálculo da circularidade média dos olhos, de forma ponderada, segundo a equação 7.1.

$$circularidade = \frac{\sum_{i=0}^n A_i \times \frac{4\pi A_i}{P_i^2}}{\sum_{i=0}^n A_i} \quad (7.1)$$

onde A_i e P_i são a área e o perímetro, respectivamente, do olho i .

De modo a que o cálculo da circularidade média, bem como o tamanho médio dos olhos não fosse afectado por olhos demasiado pequenos, que no limite até poderão ser considerados ruído, foram eliminados os olhos cuja área é inferior a 5 pixels.

Posto isto, foi implementada uma forma de automatizar a extracção de todas estas características, para a totalidade das imagens que compõem o *corpus*. Os dados registados para cada imagem, são apresentados na tabela C.1.

Além das características já referidas, foram ainda recolhidas outras informações acerca dos olhos. Foram implementados métodos de contagem de sequências de pixels correspondentes aos olhos. Esta contagem foi efectuada em três direcções, ou seja, vertical, horizontal e diagonal.

Para cada uma das direcções foram efectuadas quatro contagens:

- Sequências de N pixels consecutivos (superiores ou iguais a N);
- Número de sequências de N pixels consecutivos (superiores ou iguais a N);

- Sequências de N pixels consecutivos (inferiores a N);
- Número de sequências de N pixels consecutivos (inferiores a N).

A decisão sobre um valor para N , com significado nas três direcções, teve por base um estudo onde foi efectuada a contagem nas três direcções do número de pixels consecutivos correspondentes aos olhos do queijo. Estes valores foram agrupados em intervalos de classe com dimensão 5, ou seja, as sequências de pixels com valores de 1 a 5, são somadas e agrupadas no conjunto de 0 a 5; as sequências com valores de 6 a 10, são somadas e agrupadas no conjunto de 6 a 10, e assim sucessivamente. Estes conjuntos, serão posteriormente representados em forma de histograma, tal como apresentado nas figuras 7.2a e 7.2b.

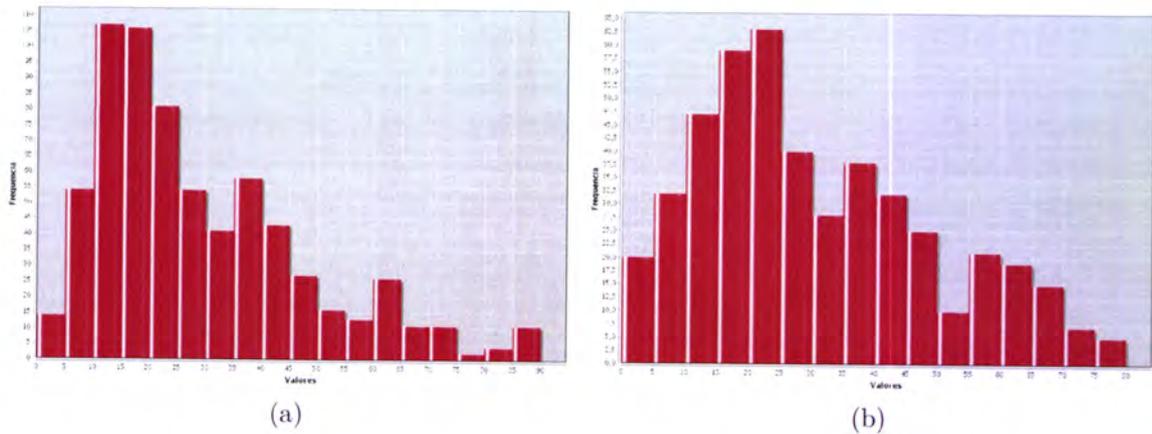
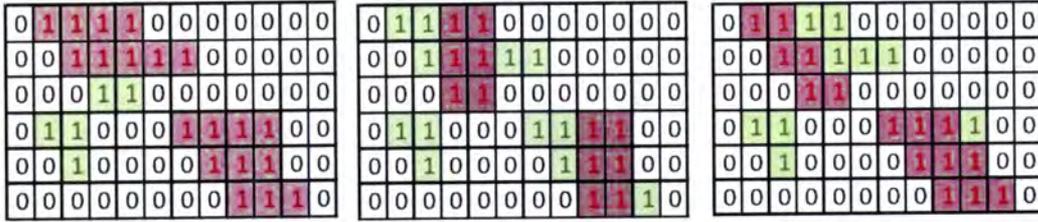


Figura 7.2: Análise da distribuição do tamanho dos olhos.

Este estudo foi efectuada para todas as imagens do *corpus* e ao serem analisados os respectivos histogramas, concluiu-se que um valor aceitável para N seria 40, visto ser o valor central na maioria dos histogramas, assim como considerar olhos com uma área superior a 1600 pixels, parece ser um pouco exagerado, considerando que o tamanho médio dos olhos aproxima-se dos 340 pixels.

Na figura 7.3 é exemplificada a contagem nas três direcções, para $N = 3$, onde os valores iguais a 1 correspondem aos pixels referentes aos olhos. A vermelho encontram-se assinaladas as sequências para $N \geq 3$ e a verde as sequências para $N < 3$. A tabela 7.1 resume os valores correspondentes às contagens.



(a) Contagem na horizontal. (b) Contagem na vertical. (c) Contagem na diagonal.

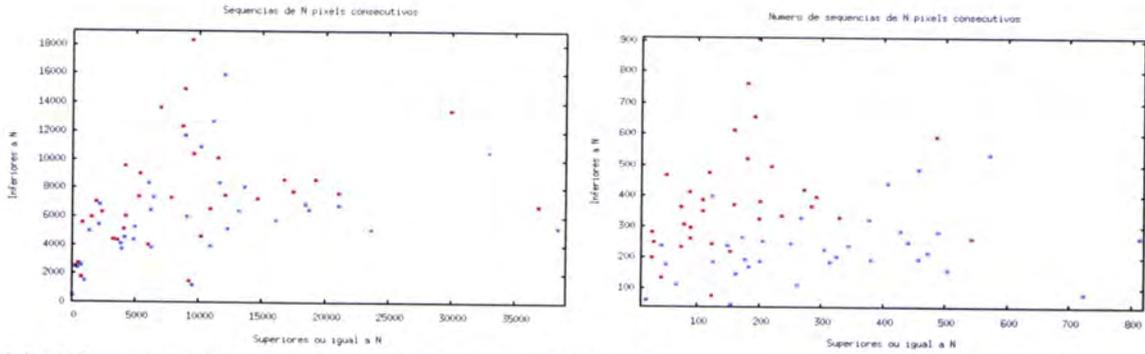
Figura 7.3: Exemplificação das direcções de contagem referentes às características dos olhos.

Tabela 7.1: Valores da contagem dos pixels referentes aos olhos, nas três direcções, para os exemplos da figura 7.3.

Direcção	Sequências de pixels		Número de sequências	
	≥ 3	< 3	≥ 3 pixels	< 3 pixels
Horizontal	19	5	5	3
Vertical	12	12	4	9
Diagonal	15	9	5	6

Uma vez que o processamento aplicado a cada imagem, descrito nos capítulos anteriores, é efectuado apenas em 1D, ou seja, na direcção horizontal, foram testados outros valores de N para esta direcção, com o objectivo de encontrar uma distribuição que permitisse que cerca de metade dos queijos tivessem sequências de N pixels consecutivos superiores ou iguais a N e a outra metade de sequências de N pixels consecutivos inferiores a N , o que se observa no gráfico 7.4a. Foi igualmente determinado um valor de N , para o qual o número de sequências de N pixels dos queijos também se encontram quantitativamente distribuídos, tal como o gráfico 7.4b.

Foi possível encontrar esta distribuição para valores de N iguais a 35 e 23, respectivamente. Nos gráficos da figura 7.4, a distribuição a vermelho corresponde a $N = 40$ e a azul as distribuições para $N = 35$ (figura 7.4a) e $N = 23$ (figura 7.4b).



(a) Número total de pixels em seqüências de N pixels consecutivos. (b) Número de seqüências de N pixels consecutivos.

Figura 7.4: Gráficos de dispersão que demonstram a relação entre olhos pequenos e olhos grandes.

7.2.1 Contagem na diagonal

A contagem nas direcções horizontal e vertical, é efectuada de forma intuitiva, correspondendo a percorrer as linhas e colunas da matriz da imagem, respectivamente.

Já no caso da direcção diagonal, o procedimento não é tão natural, visto que não é exigido que a matriz da imagem seja uma matriz quadrada, pelo que foi implementado um algoritmo que prevê o caso em que o número de linhas e colunas da matriz não é igual. O pseudo-código seguinte ilustra a solução implementada.

Algorithm 7 Contagem na Diagonal

```

1: procedure CONTADIAGONAL( $I(i, j)$ )
2:    $colunas \leftarrow Colunas(I(i, j))$ 
3:    $linhas \leftarrow Linhas(I(i, j))$ 
4:    $NumeroDiagonais \leftarrow colunas + linhas - 1$ 
5:    $c \leftarrow 0$ 
6:    $l \leftarrow 0$ 
7:    $cAnterior \leftarrow 0$ 
8:    $lAnterior \leftarrow 0$ 
9:    $triangulo \leftarrow 0$   $\triangleright 1 - TrianguloInferior; 0 - TrianguloSuperior$ 
10:  while  $NumeroDiagonais > 0$  do
11:    if  $triangulo == 1$  then
12:      while  $c < colunas \wedge l < linhas \wedge triangulo == 1$  do
13:         $CONTA()$ 
14:         $l = l + 1$ 
15:         $c = c + 1$ 
16:      end while
17:       $REINICIA\_CONTADOR()$ 
18:       $l = lAnterior + 1$ 
19:       $c = cAnterior$ 
20:      if  $lAnterior == linhas || l == linhas$  then
21:         $triangulo = 0$ 
22:         $l = 0$ 
23:         $c = 1$ 
24:         $lAnterior = 0$ 
25:         $cAnterior = 0$ 
26:      end if
27:       $lAnterior = lAnterior + 1$ 
28:    else if  $triangulo == 0$  then
29:      while  $c < colunas \wedge l < linhas \wedge triangulo == 0$  do
30:         $CONTA()$ 
31:         $l = l + 1$ 
32:         $c = c + 1$ 
33:      end while
34:       $c = cAnterior + 1$ 
35:       $l = 0$ 
36:      if  $cAnterior == colunas$  then
37:         $Interrompe$ 
38:      end if
39:    end if
40:     $NumeroDiagonais = NumeroDiagonais + 1$ 
41:  end while
42:  return  $Vector[Olhos]$ 
43: end procedure

```

7.3 Atributos relativos à cor

No que se refere aos atributos de cor, foram implementados todos os métodos que permitem a sua extracção de forma automática.

Com estes parâmetros pretende avaliar-se:

- a homogeneidade da cor do queijo revelando um estado de maior ou menor cura;
- se um queijo se encontra demasiado branco, ou demasiado escuro, denotando ausência ou excesso de cura, respectivamente.

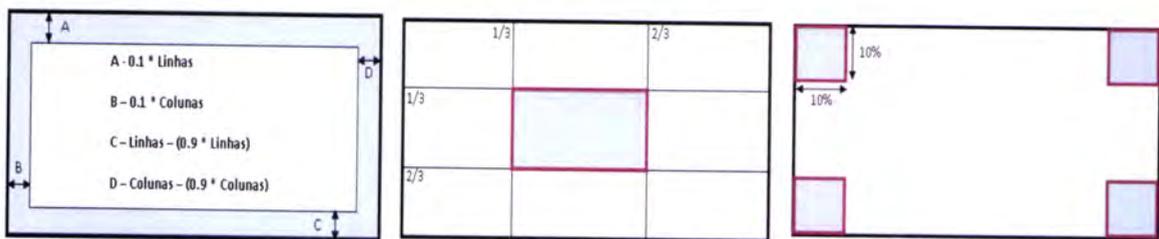
Tendo em vista esta avaliação, as zonas do queijo foram divididas em três regiões:

Crosta Região correspondente a 10% do queijo, nos seus limites exteriores, tal como é representado na figura 7.5a.

Centro Na delimitação do centro do queijo, a imagem foi representada como uma matriz $Centro_{i,j} = 3 \times 3$, sendo $Centro_{1,1}$ o centro do queijo, 7.5b.

Cantos Os quatro cantos do queijo, foram definidos seguindo um raciocínio análogo ao do centro, sendo a imagem dividida numa matriz $Cantos_{i,j} = 10 \times 10$, pelo que $Cantos_{0,0}$ é o canto superior esquerdo e $Cantos_{0,9}$ o superior direito, $Cantos_{9,0}$ é o canto inferior esquerdo e $Cantos_{9,9}$ o inferior direito. A imagem 7.5c ilustra a definição dos quatro cantos do queijo.

Este procedimento é efectuado em imagens resultantes da segmentação e às quais são retiradas as zonas classificadas como olhos, ficando-se com uma imagem que se aproxima do fundo do queijo, e por isso sem a influência dos olhos.



(a) A azul a zona correspondente à crosta do queijo.

(b) A azul a zona central do queijo.

(c) A azul os quatro cantos do queijo.

Figura 7.5: Definição das regiões onde serão recolhidos parâmetros de cor.

Para cada uma das regiões definidas atrás, são extraídas diversas características relativas à cor, nomeadamente:

- Média de cada uma das componentes da cor no espaço de cores RGB;

- Média de cada uma das componentes da cor no espaço de cores HSV;
- Média na escala de cinzentos (*grayscale*)².

Com base nas características de cor anteriores, são obtidas as seguintes características:

- Diferença entre as coordenadas de cor RGB, HSV e *grayscale* do centro e da crosta;
- Diferença máxima entre o centro e cada um dos cantos, para as coordenadas RGB, HSV e *grayscale*;
- Diferença máxima entre todos os pontos de medida (centro e os 4 cantos) para as coordenadas RGB, HSV e *grayscale*.

Além destas características de zonas pontuais da imagem, é também calculada para toda a imagem correspondente ao fundo ao queijo:

- A média das componentes RGB;
- A média das componentes HSV.

No apêndice C.1, encontra-se uma breve descrição de todas as características enumeradas anteriormente.

²A conversão do espaço de cores RGB para a escala de cinzentos (*grayscale*), é efectuada tendo por base os seguintes pesos, P , para cada uma das componentes:

$$P_R = 0.299 \quad P_G = 0.587 \quad P_B = 0.114$$

Capítulo 8

Aprendizagem e Classificação

"Pattern recognition is a process of categorizing any sample of measured or observed data as a member of one of the several classes or categories. (...) now expected to discover the mechanism of their recognition, simulate it, and put it into action with the modern technology to benefit the human beings."

Sing-Tze Bow [9].

8.1 Introdução

Tendo em vista a avaliação de um queijo, segundo os parâmetros apresentados no capítulo 7, recorreu-se a técnicas de Mineração de Dados, por forma a desenvolver um modelo de aprendizagem e classificação, que será apresentado neste capítulo.

8.2 Aprendizagem

O conceito de Mineração de Dados é definido em [18], *"Data mining is the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner"*, ou seja, consiste na análise de grandes quantidades de dados, com o objectivo de descobrir relações entre si, que possam ser úteis e compreensíveis para o utilizador.

Esta pesquisa de relações entre os dados, origina a definição de um modelo que representa o conhecimento extraído do conjunto dos dados pelo algoritmo de mineração. O modelo pode tomar diversas formas, que dependem do modo como o conhecimento é representado, sendo as equações, grafos, árvores ou regras, as formas mais comuns.

Desde os primórdios dos computadores, que se investiga e desenvolve metodologias que permitam a que os computadores, ao interagirem com o ambiente que os rodeia, possam "aprender" dotando-se de uma certa autonomia. Em [43], o conceito de "Aprendizagem

Automática” é relacionado com os conceitos de «experiência», «tarefa» e «desempenho», ou seja:

Definição 1. *Um computador aprende com a experiência E , num conjunto de tarefas T , e dada uma medida de desempenho D , se o seu desempenho nas tarefas contidas em T , medido por D , melhora com a Experiência E .*

No âmbito da Aprendizagem Automática, são comumente descritos três paradigmas:

Aprendizagem Supervisionada A aprendizagem é realizada com exemplos de treino compostos por um vector de características e o valor desejado à saída. Este procedimento assemelha-se à presença de um “professor” que para o exemplo X indica a saída Y .

Aprendizagem por Reforço Existe igualmente a presença de um “professor”, embora este não indique a resposta correcta. O “professor” apenas se limita a acompanhar o processo de aprendizagem por parte do algoritmo de aprendizagem, atribuindo avaliações positivas ou negativas mediante a resposta do modelo a determinado exemplo.

Aprendizagem Não Supervisionada Não é indicada nenhuma avaliação de cada exemplo *a priori*. A aprendizagem baseia-se na descoberta de relações entre as características dos dados de entrada fornecidos ao modelo.

Ao longo dos tempos, têm vindo a ser desenvolvidos diversos algoritmos de aprendizagem e classificação, com capacidade de se adaptarem às características específicas de determinados tipos de problemas. De entre os conjuntos de algoritmos mais conhecidos, destacam-se as árvores de decisão, regras de classificação, redes neuronais artificiais e máquinas de suporte vectorial (*support vector machines*, SVM).

Tendo em consideração que o problema em causa nesta tese é um problema de classificação, foram utilizados essencialmente algoritmos baseados em árvores de decisão, visto que se adaptam bastante bem a este tipo de problemas [43].

As árvores de decisão são um modelo bastante utilizado na área da aprendizagem automática uma vez que constroem um modelo de fácil compreensão e interpretação para qualquer ser humano [43]. A forma de aprendizagem utilizada nas árvores de decisão assenta num processo recursivo desde a raiz até cada uma das folhas da árvore, sendo que a cada exemplo do conjunto de treino é atribuído uma classificação. Cada nó da árvore está relacionado com um atributo a ser testado, cada ramo baseia-se num subconjunto dos valores dos exemplos para esse atributo, estando em cada folha uma das possíveis classificações para cada exemplo.

Ao ser apresentado um novo exemplo à árvore de decisão, é testado o atributo correspondente à raiz da árvore, seguindo depois pelo ramo correspondente ao valor do atributo. Este processo é repetido recursivamente até que um nó terminal ou folha seja alcançado. A classe presente na folha, é a classificação prevista para o exemplo.

8.3 Avaliação do especialista

Para treino e teste do classificador foram utilizadas 34 fotografias do interior dos queijos, recolhidas ao longo de diversas provas de queijos. Estas fotografias foram previamente classificadas por um especialista, a Eng. Maria da Graça Machado, responsável pelo Laboratório de Tecnologia e Qualidade dos Produtos Regionais, na Universidade de Évora.

Esta avaliação foi efectuada tendo em consideração apenas as características avaliáveis através do processamento de imagens, apresentadas no capítulo 7. As 34 fotografias foram assim, divididas em 6 classes:

1. Muito Bom
2. Bom
3. Suficiente
4. Insuficiente
5. Mau - Escuro
6. Mau - Branco

Tal como foi referido no capítulo introdutório desta tese, o interior de um queijo deve apresentar-se com a pasta fechada e bem ligada, com aspecto untuoso e com alguns olhos pequenos, cor amarelada e uniforme. No caso de um queijo ser bastante escuro ou bastante branco, é um factor que só por si atribui uma classificação de Mau, pelo que numa fase posterior as classes 5 e 6, da lista anterior, foram agrupadas numa única classe, denominada por Mau.

No esquema da figura 8.1, é apresentada a classificação atribuída a cada queijo pela Eng. Maria da Graça Machado. As instâncias de transição indicam casos em que a classificação é intermédia entre duas classes.

Às instâncias de transição indicadas na figura 8.1, com os números 2 e 32 foi atribuída a classificação de Bom, a instância 9 foi classificada como Muito Bom, a instância 10 foi classificada como Suficiente e, por sua vez, a instância 27 classificada como Mau-Branco.

8.4 Modelo de aprendizagem

A construção do modelo de aprendizagem e classificação, teve por base a escolha de um algoritmo ou conjunto de algoritmos disponíveis na ferramenta de mineração de dados, Weka [73], que com as características recolhidas em cada imagem, anteriormente apresentadas no capítulo 7, permite a definição de um modelo.

A primeira abordagem, embora um pouco *naïve*, consistiu em testar todos os algoritmos disponíveis no software Weka, para o *corpus* das fotografias de teste, com o conjunto

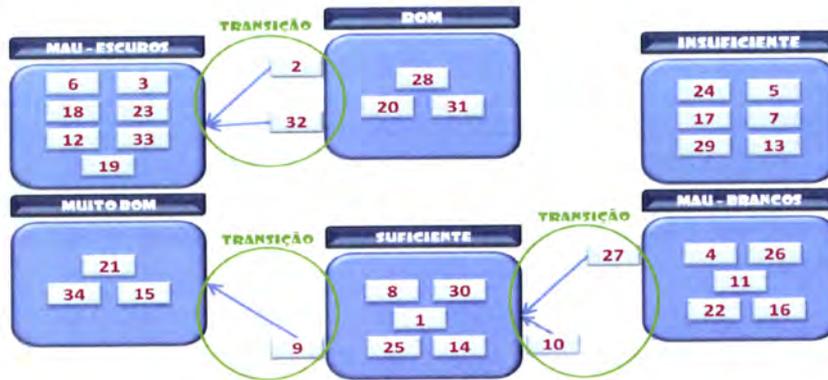


Figura 8.1: Definição e atribuição de classe a cada queijo do *corpus* e instâncias de transição.

de todas as características de cada imagem. Nenhum dos algoritmos testados apresentou uma percentagem considerável de instâncias correctamente classificadas, pelo que esta abordagem foi abortada, no entanto, estes testes revelaram-se úteis para compreender quais as características com maior relevância para o processo de aprendizagem e classificação.

Partindo do princípio que foram recolhidas características relativas aos olhos e à cor do queijo, optou-se por uma abordagem que permitisse a combinação de diversos algoritmos de classificação, de modo a que o processo de aprendizagem assentasse em diversos parâmetros do queijo a serem considerados individualmente. Cada queijo é, então, avaliado relativamente aos olhos, à cor na sua generalidade, à homogeneidade comparando o centro e a crosta e, finalmente, por uma combinação de características dos olhos e da cor. A avaliação final será por isso, apoiada nestes quatro parâmetros de classificação intermédios.

A abordagem seguida neste trabalho e que se fundamenta na manipulação dos exemplos de treino com o objectivo de gerar múltiplos modelos, foi desenvolvida nos finais do século *XX* e tem por base a capacidade de combinar diversos algoritmos de aprendizagem, obtendo-se por vezes resultados surpreendentes [73], [43]. De entre os esquemas desenvolvidos, destacam-se os esquemas designados por *voting*, *bagging*, *boosting* e *stacking* [70].

O algoritmo apresentado em [10], denominado por *bagging*, permite gerar diversos modelos, usando-os para obter uma classificação agregada. Os diversos modelos são obtidos pelo método de *bootstrap*¹, em que cada modelo é formado por uma amostra de exemplos de treino, escolhidos aleatoriamente do conjunto inicial, onde pode ocorrer a repetição de exemplos. Após a formação dos vários modelos, estes são classificados, sendo no final realizada uma média das classificações de cada amostra, pelo processo de votação (*voting*). A classificação final, resultará então de uma média das classificações intermédias, obtidas por este processo.

¹Método de divisão do conjunto de exemplos, baseado num procedimento estatístico de amostragem com substituição. Do conjunto inicial são escolhidos aleatoriamente os exemplos de treino, mas com a possibilidade de ocorrerem repetições. Os exemplos de teste são os restantes, ou seja, os que não foram incluído no conjunto de treino [43].

O algoritmo de *boosting* [57], tal como o algoritmo *bagging*, gera modelos por amostragem com substituição, no entanto permite uma ponderação do peso de cada exemplo. No início do processo de aprendizagem, todos os exemplos têm o mesmo peso. O erro é calculado em função dos exemplos classificados incorrectamente, pois estes passam a ter maior peso que os exemplos classificados correctamente. Este método embora permita uma boa percentagem de exemplos correctamente classificados, é muito susceptível à presença de ruído nos dados, o que piora bastante os resultados [76].

O esquema de *stacking*, ao contrário dos esquemas de *bagging* e *boosting*, permite combinar diferentes tipos de algoritmos de aprendizagem [73], pelo que, este esquema perfila-se como um esquema adequado ao problema em causa, visto não impor restrições ao tipo de algoritmo a utilizar, dando total liberdade para a escolha do algoritmo que melhor se adequa ao parâmetro que se pretende analisar.

8.5 Modelo de classificação

8.5.1 Algoritmo de *Stacking*

Na sequência do que foi referido na secção anterior, o esquema de *stacking* constitui uma alternativa à combinação de diversos métodos de classificação pelo método de votação, utilizada no esquema de *bagging*. A utilização do método de votação apenas faz sentido se os algoritmos de aprendizagem, sobre os quais recai a votação, forem comparáveis [73]. O desenvolvimento do método de *stacking* deu origem ao conceito de *metalearner*, que substitui o método de votação.

O esquema de *stacking* tenta aprender usando um algoritmo, o *metalearner*, que descobre a melhor forma de combinar o resultado dos algoritmos base [73]. Este processo é realizado em dois níveis, sendo que no *nível-0* são geradas as predições utilizadas como *input* para o *meta-modelo* do *nível-1*.

O conceito de dividir o processo de aprendizagem em dois níveis, foi o escolhido para construir o classificador que permitirá avaliar um queijo, através das características descritas no capítulo 7.

No *nível-0*, cada queijo é avaliado segundo 4 parâmetros, com 4 algoritmos de aprendizagem diferentes:

- Cor geral do queijo;
- Homogeneidade da cor;
- Olhos e suas características;
- Combinação de características dos olhos e da cor.

Para o nível seguinte, são utilizadas as classificações obtidas no *nível-0*, que ao serem introduzidas num outro algoritmo de aprendizagem, este combina as 4 classificações anteriores, produzindo a classificação final do queijo.

8.5.2 *Nível-0*

8.5.2.1 Classificação em função da cor - Algoritmo *Sequential Minimal Optimization*

O primeiro parâmetro a avaliar é a cor geral do queijo. Para tal foi utilizado o seguinte conjunto de características:

- Média R Centro
- Média G Centro
- Média B Centro
- Média Gray Centro
- Média R Crosta
- Média G Crosta
- Média B Crosta
- Média Gray Crosta
- Diferença Centro Crosta R
- Diferença Centro Crosta G
- Diferença Centro Crosta B
- Diferença Centro Crosta Gray
- Média Fundo H
- Média Fundo S
- Média Fundo V

Nesta classificação os queijos são classificados tendo em consideração 4 classes:

Branco Queijo globalmente claro, denunciando muito pouco tempo de cura;

Escuro Queijo globalmente escuro, denunciando demasiado tempo de cura;

Centro Escuro - Centro e crosta curados, estando conforme o esperado;

Centro Claro - Crosta curada, mas centro pouco curado.

Foram testados todos os algoritmos disponíveis no Weka, sendo o algoritmo SMO (*Sequential Minimal Optimization*) o que apresenta um melhor resultado em termos de percentagem de instâncias correctamente classificadas. O algoritmo SMO, apresentado em [50], foi desenvolvido tendo como objectivo o treino de SVMs.

As *Support Vector Machines*, desenvolvidas por Vapnik [16], em 1995, é uma técnica de Aprendizagem Máquina (AM) amplamente utilizada em diversos domínios, desde a classificação de textos à análise de imagens. As SVMs são suportadas pela teoria da aprendizagem estatística, consistindo num conjunto de princípios a ser seguidos de modo a obter-se um classificador com boa capacidade de generalização.

Dependendo do tipo de problema em análise, as SVMs podem ser de dois tipos, lineares ou não lineares. SVMs lineares são usadas para a obtenção de fronteiras lineares para a separação dos dados em duas ou mais classes, dependendo do problema em análise. Para problemas não lineares, são utilizadas SVMs não lineares, que permitem o mapeamento do conjunto de treino do seu espaço original para um espaço de maior dimensão, fazendo com que o conjunto mapeado possa ser separado por uma SVM linear [6].

Em [49], as SVMs são apresentadas como um problema de optimização, em que uma elevada complexidade do problema em causa, pode impossibilitar a aplicação dos métodos tradicionais de optimização, pelo que a decomposição do problema inicial em sub-problemas, tem sido uma abordagem seguida por diversos autores para o treino de SVMs.

Um tipo especial de decomposição, é a utilização do método SMO, onde é considerado um pequeno conjunto de variáveis para cada sub-problema. No caso concreto deste método, são consideradas duas variáveis de cada vez, evitando a necessidade da aplicação de técnicas de optimização.

Este algoritmo, quando aplicado à classificação do queijo quanto à cor, permite uma percentagem de instâncias correctamente classificadas na ordem dos 94%, cometendo apenas dois erros, no conjunto das 34 fotografias utilizadas para treino do classificador. Estes resultados foram obtidos por validação cruzada².

A figura 8.2 resume os resultados, apresentando a matriz de confusão³ do algoritmo.

8.5.2.2 Classificação em função dos olhos - Algoritmo J48

A avaliação do queijo relativamente ao número, forma e dimensão dos seus olhos, é efectuada mediante as seguintes características:

²O conjunto dos exemplos de treino, é dividido em conjuntos mutuamente exclusivos, aplicando cada modelo alternadamente a cada conjunto.

³Matriz de confusão é a matriz $M = N \times N$ em que cada elemento M_{ij} indica quantos exemplos do conjunto de treino da classe j foram classificados com a classe i .

Cor - SMO			
Instâncias Classificadas Correctamente	32	94.1176%	
Instâncias Classificadas Erradamente	2	5.8824%	
Matriz de Confusão			
a	b	c	d ← classificação
5	0	0	1 a = Branco
0	8	0	0 b = Escuro
0	0	14	0 c = CentroEscuro
0	0	1	5 d = CentroClaro

Figura 8.2: Matriz de confusão do algoritmo *Sequential Minimal Optimization*.

- Número de Olhos
- Tamanho Médio
- Fracção Área
- Circularidade

Relativamente aos olhos, os queijos são classificados em 3 classes:

Muitos - Elevada área ocupada pelos olhos, o que corresponde a um tempo de cura insuficiente;

Médios - Os olhos ocupam uma área aceitável, sendo um queijo possivelmente razoável;

Poucos - Reduzida área ocupada pelos olhos, pelo que, no que aos olhos diz respeito, o queijo poderá ser classificado como bom.

Grande parte dos algoritmos de classificação baseados em árvores de decisão realizam uma pesquisa sobre o espaço de árvores possíveis, segundo uma heurística descendente [43], sendo que o algoritmo ID3 implementa esta abordagem.

Este algoritmo constrói a árvore de decisão de forma recursiva e descendente. O atributo a testar na raiz da árvore, é obtido por um teste estatístico, que determina a capacidade de cada atributo por si só de classificar os exemplos dados. Cada valor possível para o atributo seleccionado para a raiz, dará origem a um ramo na árvore, sendo posteriormente os exemplos de treino distribuídos pelo ramo correspondente.

O procedimento é repetido para cada ramo, utilizando o subconjunto de exemplos de treino correspondente para determinar o melhor atributo a testar nesse ramo, excepto o

atributo seleccionado para a raiz. Este processo termina, quando não existem mais atributos a testar, ou quando todos os exemplos de determinado ramo, têm o mesmo valor para a classe, sendo esta a classe prevista.

Os algoritmos de indução em árvores de decisão podem provocar o sobre-ajustamento⁴ dos dados, pelo que em 1993, Quinlan apresentou um versão modificada do algoritmo ID3, denominado por C4.5, utilizando a técnica de poda de regras⁵ de forma a minimizar os efeitos do sobre-ajustamento.

No software Weka, encontra-se o algoritmo C4.5 renomeado para J48 [11], onde se encontra implementada uma forma de poda da árvore gerada pelo algoritmo ID3.

Para a avaliação do queijo com as características respeitantes aos olhos, o algoritmo J48 apresentou os melhores resultados, de entre os algoritmos testados.

A figura 8.3 apresenta os resultados do algoritmo J48, para classificar um queijo relativamente às características dos olhos.

Olhos - J48		
Instâncias Classificadas Correctamente	32	94.1176%
Instâncias Classificadas Erradamente	2	5.8824%
Matriz de Confusão		
a	b	c <-- classificação
11	0	0 a = Muitos
0	7	0 b = Medios
1	1	14 c = Poucos

Figura 8.3: Matriz de confusão do algoritmo J48.

8.5.2.3 Classificação em função da homogeneidade da cor - Algoritmo *Best-First Decision Tree*

A homogeneidade da cor do queijo, é avaliada mediante as características de cor do centro, da crosta e dos cantos do queijo, e por características resultantes da comparação entre o centro, a crosta e os cantos, formando o seguinte conjunto:

- Média R Centro

⁴O sobre-ajustamento ocorre quando se foca excessivamente nos exemplos de treino, provocando a diminuição da capacidade de generalização do modelo [56].

⁵Transformação da árvore de decisão num conjunto de regras de classificação.

- Média G Centro
- Média B Centro
- Média Gray Centro
- Média R Crosta
- Média G Crosta
- Média B Crosta
- Média Gray Crosta
- Diferença Centro Crosta R
- Diferença Centro Crosta G
- Diferença Centro Crosta B
- Diferença Centro Crosta Gray
- Média H Centro
- Média S Centro
- Média V Centro
- Média H Crosta
- Média S Crosta
- Média Y Crosta
- Diferença Centro Crosta H
- Diferença Centro Crosta S
- Diferença Centro Crosta V
- Média R Canto Sup Esquerda
- Média G Canto Sup Esquerda
- Média B Canto Sup Esquerda
- Média Gray Canto Sup Esquerda

- Média R Canto Sup Direita
- Média G Canto Sup Direita
- Média B Canto Sup Direita
- Média Gray Canto Sup Dir
- Média R Canto Inferior Esquerda
- Média G Canto Inferior Esquerda
- Média B Canto Inferior Esquerda
- Média Gray Canto Inferior Esquerda
- Média R Canto Inferior Direita
- Média G Canto Inferior Direita
- Média B Canto Inferior Direita
- Média Gray Canto Inferior Direita
- Máximo Centro Cantos R
- Máximo Centro Cantos G
- Máximo Centro Cantos B
- Máximo Centro Cantos Gray
- Máximo Centro Cantos Todos R
- Máximo Centro Cantos Todos G
- Máximo Centro Cantos Todos B
- Máximo Centro Cantos Todos Gray

Um queijo classificado quanto à homogeneidade da sua cor, poderá corresponder a uma de duas classes:

Sim - Verifica-se a presença de uma cor homogénea;

Não - A cor não se apresenta homogénea.

O tipo de árvores de decisão usado para a classificação do queijo em função da homogeneidade da sua cor, apresenta as características padrão de uma árvore de decisão, referidas na introdução deste capítulo.

No entanto, o algoritmo de árvores de decisão BFTree apresenta uma diferença. Enquanto nas árvores de decisão tradicionais expandem os seus nós em profundidade, as árvores BFTree expandem primeiro o melhor nó. Tal facto poderá não provocar diferenças nas árvores geradas, a não ser que seja pré-definido um valor para o número de expansões, e aí sim, as árvores geradas serão diferentes [60].

A figura 8.4 apresenta os resultados do algoritmo BFTree.

<u>Homogeneidade - BFTree</u>		
Instâncias Classificadas Correctamente	32	94.1176%
Instâncias Classificadas Erradamente	2	5.8824%
Matriz de Confusão		
a b <-- classificação		
17	1	a = Sim
1	15	b = Nao

Figura 8.4: Matriz de confusão do algoritmo *Best-First Decision Tree*.

8.5.2.4 Classificação em função da cor e dos olhos - Algoritmo *Alternating Decision Tree*

Por fim, o último parâmetro a avaliar, é uma combinação da avaliação relativa aos olhos e à cor, mas que permite uma boa separação dos queijos bons dos maus. Portanto, o conjunto de características utilizadas nesta avaliação, contém características da cor e dos olhos:

- Número de Olhos
- Tamanho Médio
- Fracção Área
- Circularidade
- Média B Centro
- Média Gray Centro

- Média B Crosta
- Média Gray Crosta
- Diferença Centro Crosta B
- Diferença Centro Crosta Gray
- Máximo Centro Cantos B
- Máximo Centro Cantos Gray
- Máximo Centro Cantos Todos B
- Máximo Centro Cantos Todos Gray
- Média Fundo B

Nesta classificação são consideradas duas classes:

Muito Bom Os olhos são poucos e pequenos e a cor é amarelada e uniforme, respeitando os critérios do queijo de Évora, e por isso corresponde a um queijo potencialmente bom;

Mau Não verifica as condições anteriores (poucos olhos e cor amarelada e uniforme).

Para esta classificação, foi utilizado o algoritmo ADTree (*Alternating Decision Tree*), apresentado em [24], que resulta da combinação do conceito de árvores de decisão com o método de *boosting* originando regras de classificação menores, e por isso, mais fáceis de interpretar.

As *alternating decision tree* são similares às *option trees* [24], que diferem das árvores de decisão uma vez que apresentam dois tipos de nó: nós de decisão e nós de opção. Para classificar um exemplo, num nó de decisão segue-se por um dos ramos do nó, no entanto existe um nó de opção que permite seguir por todos os ramos. Isto significa que cada exemplo poderá ter mais do que uma folha, pelo que a classificação final é uma combinação das classificações de cada folha, utilizando o método de votação.

As *options trees* podem originar uma ADTree por adição incremental de nós, usando um algoritmo de *boosting*, onde os nós de decisão são designados por *splitter nodes* e os nós de opção designados por nós de predição. Este tipo de árvores está indicado para problemas de classificação de duas classes, sendo que cada nó de predição está associado a um valor numérico positivo ou negativo [73].

A figura 8.5 resume os resultados do algoritmo *Alternating Decision Tree*.

No apêndice D.1 encontra-se a tabela D.1, onde são apresentados os resultados dos diferentes classificadores, para cada um dos parâmetros em que assenta a classificação de um queijo.

<u>BonsMaus - ADTree</u>			
Instâncias Classificadas Correctamente	32	94.1176%	
Instâncias Classificadas Erradamente	2	5.8824%	
Matriz de Confusão			
a	b	-- classificação	
20	1	a = MuitoBom	
1	12	b = Mau	

Figura 8.5: Matriz de confusão do algoritmo *Alternating Decision Tree*.

8.5.3 *Nível-1*

8.5.3.1 Algoritmo *RandomTree* - Esquema de *Stacking*

O modelo do *nível-1* é definido em função dos resultados das classificações do *nível-0*, pelo que as características de *input* para o *metalearner* são, para cada parâmetro avaliado no *nível-0*:

- Cor geral
 - Branco
 - Escuro
 - Centro Escuro
 - Centro Claro
- Olhos
 - Muitos
 - Médios
 - Poucos
- Homogeneidade
 - Sim
 - Não
- Cor e Olhos

- Muito Bom
- Mau

A classificação final de um queijo, corresponderá a uma das seguintes classes:

Muito Bom - Queijo conforme a definição, ou seja, com poucos ou nenhuns olhos e cor ligeiramente amarelada;

Bom - Queijo onde se verifica a presença de alguns olhos e cuja cor poderá não ser tão uniforme quanto os queijos da classe Muito Bom;

Suficiente - Queijo com olhos de dimensões médias e cor pouco uniforme;

Insuficiente - Queijo com olhos de dimensões consideráveis e cor muito pouco uniforme;

Mau - Queijos com cor muito escura ou muito clara denotando tempo de cura incorrecto. No caso de queijos muito brancos, os olhos têm dimensões elevadas, denotando muito pouco tempo de cura.

Estas classes foram assim definidas, tendo em consideração a opinião de um especialista, a Eng. Maria da Graça Machado, tal como descrito em 8.3.

O algoritmo utilizado para realizar a aprendizagem neste nível, foi igualmente um algoritmo de árvores de decisão, denominado por RandomTree, cuja particularidade reside no facto de que em cada nó da árvore podem ser considerados K atributos diferentes, em que K é um número aleatório, não sendo aplicado o método de poda à árvore gerada [73].

Na figura 8.6 são resumidos os resultados do algoritmo RandomTree, referentes ao *nível-1* do processo de segmentação.

Stacking - RandomTree					
Instâncias Classificadas Correctamente	29	85.2941%			
Instâncias Classificadas Erradamente	5	14.7059%			
Matriz de Confusão					
a	b	c	d	e	<-- classificação
3	1	0	0	0	a = MuitoBom
1	4	0	0	0	b = Bom
0	1	5	0	0	c = Suficiente
1	0	0	5	0	d = Insuficiente
0	0	1	0	12	e = Mau

Figura 8.6: Matriz de confusão do algoritmo RandomTree.

8.5.3.2 Análise dos resultados

A forma mais simples de avaliar um modelo de classificação, consiste em contar o número de instâncias para as quais o valor previsto para a classe coincide com o valor real [43]. Esse valor, normalizado em termos de percentagem, é obtido pela soma dos valores da diagonal da matriz de confusão, figura 8.7, podendo tomar a designação de Percentagem de Instâncias Correctamente Classificadas (PICC).

		Previsto	
		Negativo	Positivo
Desejado	Negativo	a	b
	Positivo	c	d

Figura 8.7: Matriz de confusão 2×2 .

Além da métrica PICC, podem ser consideradas duas outras métricas:

Sensibilidade : $\frac{d}{c+d} \times 100(\%)$

Especificidade : $\frac{a}{a+b} \times 100(\%)$

O esquema da figura 8.8, apresenta os valores das as 3 métricas para cada uma das classes que pode ser atribuída a um queijo.

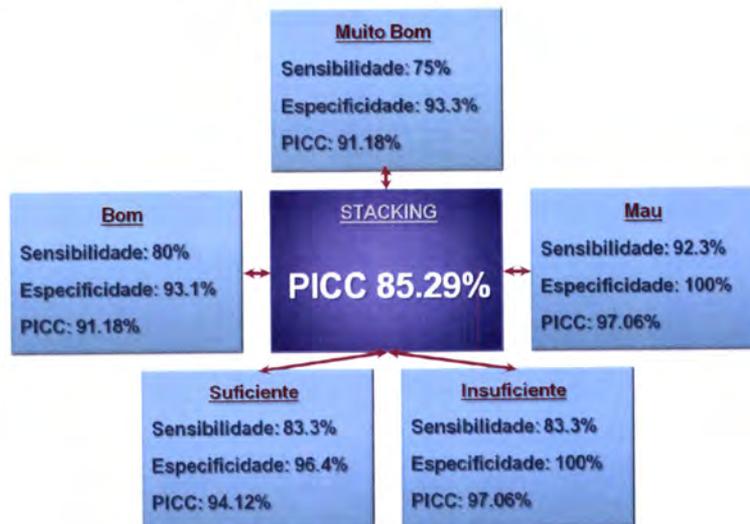


Figura 8.8: Valores das métricas de avaliação do modelo de *Stacking*.

Um classificador é tanto mais próximo do ideal quanto maiores forem as percentagens para as 3 métricas [43]. No entanto, deve ter-se em conta que existe um compromisso entre sensibilidade e especificidade, pois um aumento da sensibilidade pode diminuir a especificidade.

Tendo em conta que apenas a sensibilidade da classe Muito Bom tem uma percentagem inferior a 80%, pode-se considerar uma avaliação muito positiva do modelo implementado.

8.5.3.3 Árvore de classificação final

Na figura 8.9 é possível observar o aspecto final da árvore de decisão obtida no *nível-1* do esquema de *stacking*, a qual permite obter uma percentagem de 85.29% de novos exemplos correctamente classificados.

8.5.3.4 Resumo das classificações do *nível-1*

No apêndice D.2 encontra-se a tabela D.2, onde são apresentados os resultados da classificação final com o esquema de *stacking*, para todas as imagens que integram o *corpus*, que foi utilizado para treino e teste do classificador.

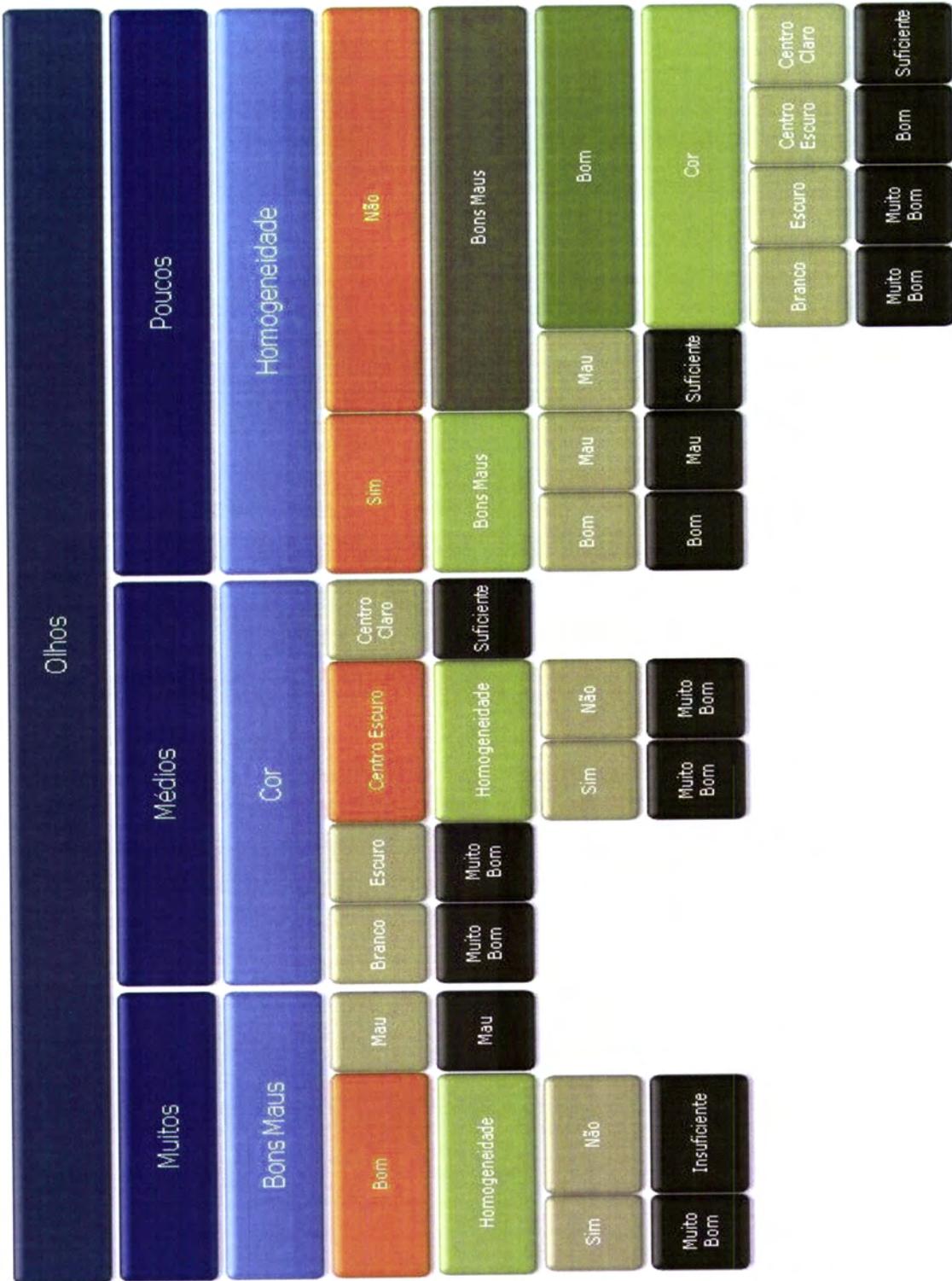


Figura 8.9: Esquema da árvore de decisão gerada pelo algoritmo de *Stacking*.

8.6 Abordagem alternativa

Na primeira abordagem para o desenvolvimento do modelo de classificação, testaram-se todos os algoritmos disponíveis na ferramenta Weka, para o conjunto de todos os atributos retirados de cada imagem (apêndice C.1).

Voltou-se a verificar que as árvores de decisão é um tipo de algoritmo com bons resultados para problemas de classificação, sendo neste caso o algoritmo *RandomForest* [72], [68], o que apresentou melhores resultados.

A figura 8.10 apresenta a matriz de confusão para do algoritmo *RandomForest*, mostrando que esta abordagem simplista conduz a um resultado insatisfatório, quando comparada com os resultados apresentados em 8.5.3.2.

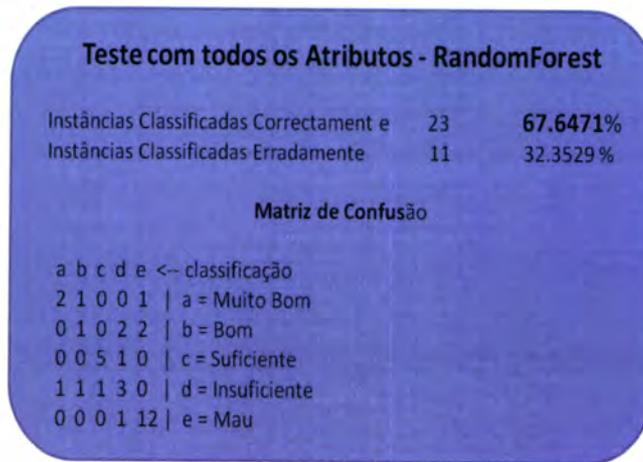


Figura 8.10: Matriz de confusão do algoritmo RandomForest.

Uma consequência natural da matriz de confusão da figura 8.10, é a diminuição das percentagens das métricas de avaliação do algoritmo, comparando com as apresentadas em 8.11.

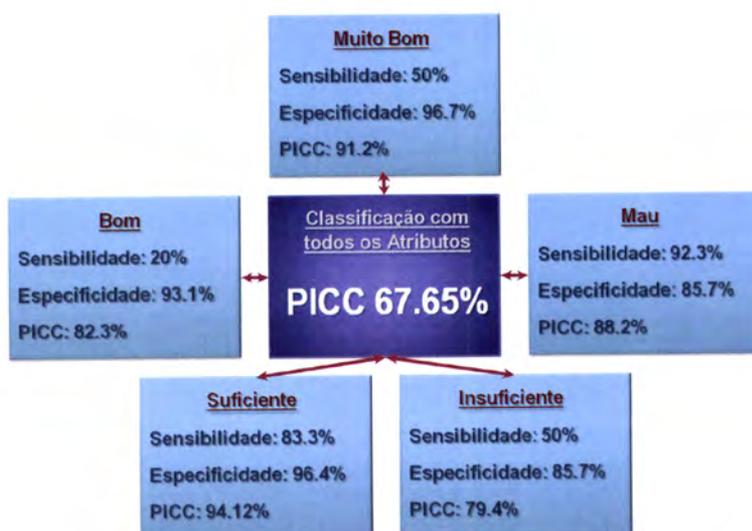


Figura 8.11: Valores das métricas de avaliação do algoritmo Random Forest.

Capítulo 9

Aplicação de Software

9.1 Introdução

Neste capítulo é apresentada a aplicação de software que foi desenvolvida, constituindo um forma de integrar todos os algoritmos apresentados ao longo desta tese, tornando a experiência para o utilizador o mais simples possível.

O código da aplicação foi implementado na totalidade com a linguagem de programação JAVA (1.6.0_03), aproveitando todas as suas potencialidades de desenvolvimento e independência da plataforma onde a aplicação irá ser executada. O ambiente de desenvolvimento consistiu no editor de texto Kate (2.5.8) e no sistema operativo Ubuntu 7.04.

Na aplicação desenvolvida, foram integrados componentes de outros programas igualmente desenvolvidos em JAVA, nomeadamente ImageJ (versão 1.42q), JFreeChart (versão 1.0.13) e Weka (versão 3.6.3).

9.2 Interface de utilizador

A interface constitui a única forma de interacção entre o utilizador e a aplicação, pelo que o seu desenvolvimento foi orientado à sua fácil usabilidade.

É disponibilizado visualmente ao utilizador, apenas as informações consideradas relevantes, o que permitiu seccionar a interface em cinco zonas principais:

- No lado esquerdo, é apresentado a lista das imagens contidas na pasta "imagens/", pasta pré-definida da aplicação, onde devem ser colocadas as imagens a processar;
- Relativamente ao centro, e em destaque, é apresentada a fotografia actualmente seleccionada pelo utilizador;
- Do lado direito e em cima, após o processamento da fotografia seleccionada, é apresentada a imagem correspondente à região do queijo da fotografia inicial, resultado do

recorte da imagem original;

- Do lado direito em baixo, é exibido o resultado da segmentação e reconhecimento dos olhos, sobre a imagem resultante do ponto anterior;
- Em baixo, são resumidos em forma de tabela, os dados resultantes da segmentação e os dados utilizados para a avaliação do queijo, assim como a respectiva classificação.



Figura 9.1: Aspecto visual da interface.

9.2.1 Modos de utilização

Tal como referido na secção anterior, as fotografias a processar pela aplicação devem ser colocadas na pasta "imagens/". Ainda assim, é dado ao utilizador a possibilidade de processar fotografias que estejam em qualquer outra localização, bastando para isso seguir no menu "Ficheiro" a opção "Abrir" e então escolher a fotografia.

No entanto, e tendo em consideração as fotografias presentes na pasta "imagens/", estas poderão ser processadas de dois modos:

Individualmente Seleccionar a fotografia a processar na lista disponível do lado esquerdo da interface, e no menu "Executar" seguir a opção "Avaliar Fotografia Seleccionada", figura 9.2a.

Em Grupo Permite processar todas as fotografias presentes na pasta "imagens/", sendo apresentados todos os resultados na tabela da parte inferior da interface. Para este tipo de processamento, basta seguir o menu "Executar" e seguir a opção "Avaliar todas as Fotografias", figura 9.2b.



(a) Avaliar fotografia seleccionada.



(b) Avaliar todas as fotografias.

Figura 9.2: Formas de avaliar as fotografias.

9.2.2 Visualização dos resultados

Os resultados são apresentados em forma de tabela na parte inferior da interface. No que diz respeito ao reconhecimento e extracção de características relativas aos olhos, é mostrado ao utilizador a informação relativa ao número de olhos, tamanho médio, fracção da área ocupada pelos olhos e a circularidade média dos mesmos, segundo a equação 7.1.

É igualmente fornecida uma estimativa das dimensões do queijo, que embora não resultem directamente do procedimento de detecção dos olhos, resultam de uma fase posterior, ou seja, do reconhecimento e recorte do queijo na fotografia inicial. Portanto, além das características acima referidas é apresentada a largura e a altura do queijo, na referida tabela, tal como na figura 9.3.

Dados da Segmentação		Dados Classificação				
Fotografia	Número de Olhos	Tamanho Médio	Fracção da Área	Circularidade	Largura	Altura
3.jpg	46,00	86,70	2,23	,03	5,20	1,90
1.jpg	50,00	138,90	4,41	,05	4,27	2,04
2.jpg	19,00	63,00	1,04	,05	3,90	1,64
4.jpg	30,00	59,93	1,29	,04	5,21	1,48

Figura 9.3: Tabela com os dados da detecção dos olhos do queijo, apresentada na interface.

Os dados gerados e analisados durante o processo de avaliação e classificação do queijo, são mostrados na tabela "Dados Classificação". Nesta tabela, além da classificação final, são também apresentadas as classificações relativas à generalidade da cor do queijo, quanto à homogeneidade da cor, quanto aos olhos e suas características e a um outro parâmetro que compreende características respeitantes aos olhos e à cor. A forma de obtenção de todos estes parâmetros foi assunto do capítulo 8. A configuração desta tabela poderá ser visualizada na figura 9.4.

9.2.3 Análise de desempenho

Tendo em consideração os diversos algoritmos apresentados ao longo desta tese, o processamento de uma imagem pode ser dividido em três fases:

Dados da Segmentação		Dados Classificação			
Fotografia	Cor	Olhos	Homogeneidade	Bons Maus	Classificação final
3.jpg	Centro Escuro	Medios	Nao	Muito Bom	Muito Bom
1.jpg	Branco	Muitos	Nao	Mau	Mau
2.jpg	Centro Escuro	Poucos	Nao	Muito Bom	Muito Bom
4.jpg	Centro Escuro	Poucos	Sim	Muito Bom	Muito Bom

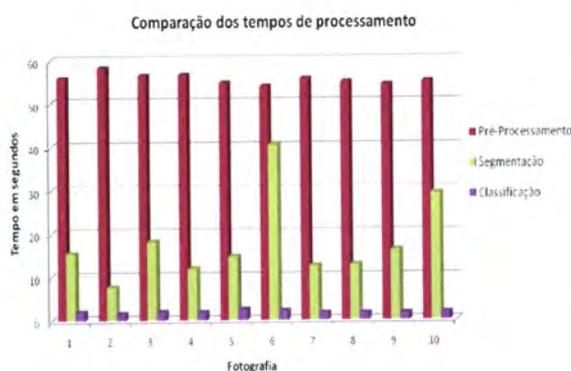
Figura 9.4: Tabela com os dados da avaliação do queijo, apresentada na interface.

Pré-processamento Inclui as etapas de detecção da moldura, eliminação de possível ruído, rotação da imagem caso se verifique um ângulo de rotação da moldura superior a 1 grau, terminando no reconhecimento do queijo;

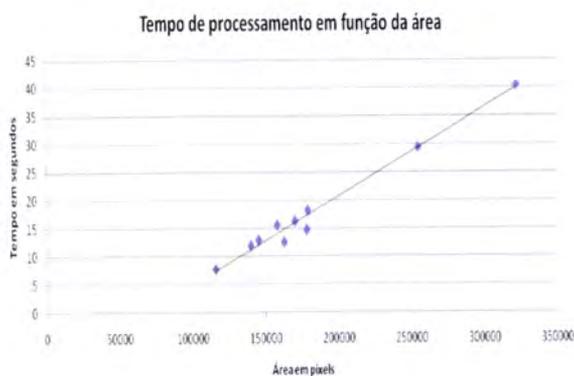
Segmentação Separação dos olhos do fundo do queijo;

Classificação Extracção de características, que mediante o modelo de classificação desenvolvido, permite a avaliação do queijo.

Foram efectuados testes com algumas fotografias experimentais, sobre as quais foram registados os tempos de processamento de cada etapa. As 10 fotografias cujos tempos se encontram resumidos no gráfico 9.5a, têm a mesma dimensão de 1536 × 2048, verificando-se que a etapa de pré-processamento é a que ocupa a maioria do tempo de processamento de uma fotografia. No gráfico de dispersão 9.5b, verifica-se, tal como seria espectável, que o tempo do processo de segmentação, varia em função da área da imagem reconhecida como queijo.



(a) Tempo de processamento de cada fase



(b) Tempo de processamento em função da área do queijo

Figura 9.5: Análise do desempenho do software.

9.2.4 Ficheiros criados

O processamento de cada fotografia, pressupõe o cumprimento de diversas etapas, abordadas ao longo desta tese. Algumas destas etapas resultam na criação de um ou mais ficheiros, que constituirão o ponto de partida da próxima etapa de processamento.

A fase de pré-processamento, capítulo 4, culmina com a criação de uma nova imagem onde apenas deve constar a região correspondente ao queijo. É sobre esta imagem que será aplicado o algoritmo de segmentação, abordado no capítulo 5, obtendo-se desta vez duas imagens. Uma das imagens identifica as regiões classificadas como olhos enquanto a outra apresenta a imagem do queijo subtraindo os olhos, constituindo uma aproximação ao fundo do queijo.

Além destas duas imagens, nesta etapa é ainda criado um ficheiro no formato *txt*, em que a cada pixel da imagem resultante da segmentação, é atribuído o valor 0 se pertencer ao fundo do queijo, caso contrário é atribuído o valor 1 indicando que pertence a um olho, tal como é exemplificado nas imagens 7.3. Com a obtenção deste ficheiro, é evitado a repetição de todo o processo de segmentação, caso seja necessário a obtenção de algum valor em particular.

O último ficheiro a ser criado, consiste num ficheiro XML onde é guardada de forma estruturada a informação relevante sobre todo o processamento realizado sobre a imagem, nomeadamente, no que se refere à detecção e contagem dos olhos, às dimensões do queijo, bem como informações relativas à avaliação do mesmo. A figura 9.6 demonstra um exemplo da estrutura do ficheiro XML, criado após a conclusão do processamento da imagem.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <queijo>
  <nomeFoto>1.jpg</nomeFoto>
  - <caracteristicasOlhos>
    <numero>50.0</numero>
    <tamanhoMedio>138.9</tamanhoMedio>
    <fracaoArea>4.408124404950809</fracaoArea>
    <circularidade>0.049492152420370764</circularidade>
  </caracteristicasOlhos>
  - <dimensoes>
    <largura>4.27</largura>
    <altura>2.03</altura>
  </dimensoes>
  - <avaliacao>
    <cor>Branco</cor>
    <olhos>Muitos</olhos>
    <homogeneidade>Nao</homogeneidade>
    <bonsMaus>Mau</bonsMaus>
    <classificacaoFinal>Mau</classificacaoFinal>
  </avaliacao>
</queijo>
```

Figura 9.6: Exemplo da estrutura do ficheiro XML gerado após o processamento e avaliação de uma fotografia.

Os ficheiros criados assim como a etapa em que tal ocorre, é resumido no esquema da figura 9.7.

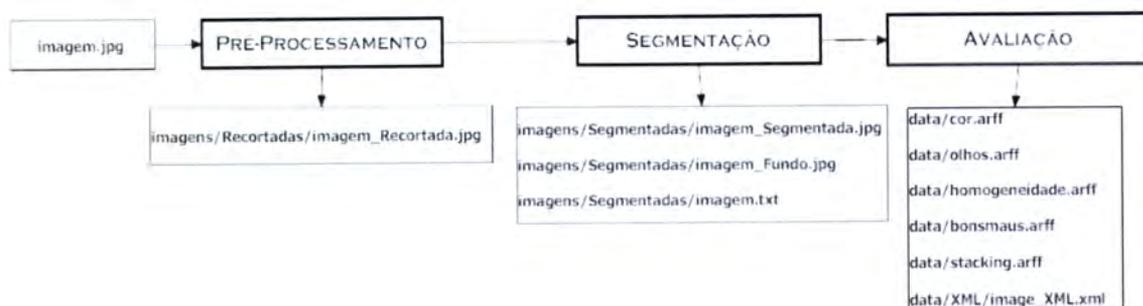


Figura 9.7: Ficheiros criados ao longo da avaliação de uma imagem.

9.2.5 Documentação

O código desenvolvido no âmbito desta aplicação de software encontra-se devidamente documentado, tendo sido utilizada a ferramenta Javadoc¹ que permite criar a documentação da API do software em formato HTML.

A figura 9.8 apresenta a descrição de cada classe² desenvolvida, com o respectivo comentário.

9.3 Software de desenvolvimento

A aplicação de software que tem vindo a ser apresentada no presente capítulo, foi desenvolvida integralmente utilizando a linguagem de programação JAVA.

Alguns dos algoritmos apresentados nos capítulos anteriores, foram implementados recorrendo ao auxílio de outros programas igualmente desenvolvidos em JAVA, que por serem *opensource* permitiram uma fácil integração no software apresentado nesta tese. O software ImageJ, foi utilizado para a análise das imagens binárias resultantes da detecção dos olhos do queijo, pelo algoritmo *Analyse Particles*.

O software Weka, muito utilizado no domínio da mineração de dados, facilitou em muito a implementação do método de classificação de um queijo, apresentado do capítulo 8.

Numa fase preliminar de estudo e compreensão do problema em causa nesta tese, foi utilizado o software JFreeChart para a obtenção e visualização de gráficos, que permitiram

¹<http://java.sun.com/j2se/javadoc/>

²O conceito de classe no âmbito da linguagem de programação JAVA, linguagem orientada a objectos, representa um objecto especial para a criação de objectos idênticos, ou seja, com estrutura interna e comportamento idênticos [42].

Class Summary	
Caracteristicas	Extracção de características do queijo
Classificador	Classe onde são criados os ficheiros .arff de input para a classificação de um queijo
DetectaMoldura	Classe com os métodos para a detecção das linhas em fotografias padronizadas Aplicação dos operadores de Prewitt
ExecutaAvaliacao	Avaliação de um queijo
ExecutaSegmentacao	Segmentação de uma imagem de queijo, para detecção dos olhos
Fotografias	Conjunto de fotografias do Corpus
Histograma	Geração de histogramas
Interface	Interface para o utilizador
LeitorXML	Classe para efectuar a leitura dos dados de classificação e segmentação guardados em formato XML.
Pixel	Classe que representa um objecto "Pixel"
Plot	Criação de gráficos de sinais, pela variação do brilho da cor da imagem ao longo de uma linha
PreProcessamento	Classe que exetuca o Preprocessamento de uma imagem
ReconheceQueijo	Classe com os métodos para o reconhecimento do Queijo em fotografias padronizadas
Rotacao	Rotação de uma imagem
Segmentacao	Classe onde são implementados os métodos necessários à detecção dos olhos do queijo.
Utilidades	Classe com métodos de auxílio utilizados por diversas classes

Figura 9.8: Documentação gerada pela ferramenta Javadoc.

a identificação de algumas das características e aspectos do problema, a ter em consideração na definição da abordagem à solução do mesmo.



Figura 9.9: Esquema que resume o software utilizado para o desenvolvimento da aplicação.

Capítulo 10

Conclusão

Neste trabalho desenvolveu-se uma aplicação de software que permite a avaliação automática de alguns parâmetros dos queijos de Évora (Denominação de Origem Protegida). A aplicação desenvolvida é uma ferramenta auxiliar que poderá ser usada tanto por membros do painel de avaliação como em controlo de qualidade pelos próprios produtores.

A aplicação analisa várias características das imagens de queijos. O número de olhos, assim como a fracção de área que ocupam, são características importantes para uma correcta avaliação da sua qualidade. Encontram-se na bibliografia diversos algoritmos e métodos de reconhecimento de objectos em imagens, no entanto, por nenhum dos algoritmos testados se adaptar ao problema de reconhecimento dos olhos do queijo, foi desenvolvido um método de segmentação adaptado ao problema, apresentado no capítulo 5, tendo sido posteriormente comparado com outros algoritmos.

Embora não tenha sido integrado no software, foi desenvolvido um método para a detecção de textura em fotografias da crosta exterior de queijos, apresentado no capítulo 6, onde a presença de textura pode indicar que o processo de secagem do queijo ocorreu demasiado depressa ou que o queijo foi pouco compactado.

O software desenvolvido foi testado com fotografias e dados reais, obtidos com a colaboração do painel de provadores do Laboratório de Tecnologia e Qualidade dos Produtos Naturais, da Universidade de Évora, obtidas e pré-processadas segundo a metodologia descrita no capítulo 4.

10.1 Perspectivas futuras

Ao longo do desenvolvimento de toda a tese, foram sendo tomadas opções que deixam em aberto a continuação do trabalho. De entre os aspectos que poderão vir a ser melhorados, destacam-se:

1. Performance do algoritmo de segmentação;

2. Generalização dos algoritmos desenvolvidos para a extracção dos atributos de cor do queijo para as merendeiras;
3. Classificação de um queijo mediante a análise de fotografias do seu interior e exterior;
4. Melhorar a capacidade de generalização do algoritmo de classificação, mediante a recolha de mais exemplos de treino.

Tendo em consideração que os filtros utilizados no algoritmo de segmentação efectuam o processamento apenas a uma dimensão (linhas da imagem), o cálculo do limiar de segmentação é igualmente independente de linha para linha, o que possibilita a paralelização do código, com o hardware adequado, melhorando a performance do algoritmo.

No caso das merendeiras ao apresentarem dimensões na ordem dos 12 a 15 cm, e por isso superiores às do queijo, é permitido que apresentem uma cor mais clara relativamente à do queijo, sendo que este aspecto terá de ser tido em consideração aquando da redefinição dos algoritmos para o processamento de imagens correspondentes a merendeiras.

No seguimento do método apresentado no capítulo 6, para a detecção de textura em imagens, fica em aberto a utilização deste método para a classificação de imagens do exterior de queijos baseada na presença ou não de textura e, por conseguinte, a classificação de um queijo pela conjugação da avaliação do seu interior e exterior.

As fotografias que constituíram o *corpus* de treino e teste do classificador, embora representem substancialmente o queijo de Évora, podem, no entanto, apresentar uma cobertura mais consistente se o conjunto de fotografias for aumentado. Com mais exemplos de treino, decerto que o classificador gerado será mais generalista e, como tal, com melhor taxa de acerto aquando na classificação de um novo exemplo.

Bibliografia

- [1] *Detection of external and internal insect infestation in wheat by near-infrared reflectance spectroscopy*, Journal of the Science of Food and Agriculture **71** (1996), no. 2, 251–264.
- [2] *Determination of live weight of pigs from dimensions measured using image analysis*, Computers and Electronics in Agriculture **15** (1996), no. 1, 57 – 72.
- [3] J. Batlle L. Pacheco J. Freixenet A. Oliver, X. Munoz, *Improving clustering algorithms for image segmentation using contour and region information*, **2** (2006), 315–320.
- [4] F. Ade, *Characterization of textures by 'eigenfilters'*, Signal Processing **5** (1983), no. 5, 451 – 457.
- [5] Ronald W. Schafer Alan V. Oppenheim, *Discrete-time signal processing*, Prentice Hall, 1998.
- [6] André C. P. L. F. de Carvalho Ana Carolina Lorena, *Uma introdução às support vector machines*, RITA **14** (2007), no. 2.
- [7] Andreas Antoniou, *Digital signal processing*, McGraw-Hill, 2006.
- [8] Hao Helen Zhang Bertrand Clarke, Esnest Fokoué, *Principles and theory for data mining and machine learning*, Springer, 2009.
- [9] Sing-Tze Bow, *Pattern recognition and image preprocessing*, Marcel Dekker, 2002.
- [10] Leo Breiman, *Bagging predictors*, Machine Learning **24** (1996), 123–140.
- [11] V.P. Bresfelean, *Analysis and predictions on students' behavior using decision trees in weka environment*, (2007), 51–56.
- [12] Ying Geng Cai-Xia Deng, Ting-Ting Bai, *Image edge detection based on wavelet transform and canny operator*, International Conference on Wavelet Analysis and Pattern Recognition, 2009, pp. 355 – 359.
- [13] Jonh Canny, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8** (1986), 679 – 698.

- [14] CERTIS, *Proposta de registo da denominação de origem do queijo de Évora*. acessível em <http://www.certis.pt/index.html> (27-julho-2010).
- [15] A.Y.S. Chia, M.K.H. Leung, How-Lung Eng, and S. Rahardja, *Ellipse detection with hough transform in one dimensional parametric space*, **5** (2007).
- [16] Corinna Cortes and Vladimir Vapnik, *Support-vector networks*, *Machine Learning* **20** (1995), 273–297.
- [17] S. Hockaday R.D. Tillett G. Ross D. Chan, N. McFarlane, *Image processing for underwater measurement of salmon biomass*, (1998), 12/1 –12/6.
- [18] Padhraic Smyth David Hand, Heikki Mannila, *Principles of data mining*, The MIT Press, 2001.
- [19] E. R. Davies, *Introduction to texture analysis*.
- [20] ———, *Image processing for the food industry*, vol. 37, World Scientific, 2000.
- [21] William E. Higgins Denis Dunn, *Optimal gabor filters for texture segmentation*, *IEEE Trasaction on imagem processing* **4** (1995), no. 7, 947–964.
- [22] Jian Yu Dongju Liu, *Otsu method and k-means*, **1** (2009), 344 –349.
- [23] T. Yambe A. Tanaka M. Yoshizawa E. dos Santos Filho, Y. Saijo, *Segmentation of calcification regions in intravascular ultrasound images by adaptive thresholding*, (2006), 446–454.
- [24] Yoav Freund and Llew Mason, *The alternating decision tree learning algorithm*.
- [25] Maiyu Zhou Zhonglin Xu Yongtian Chen Guang Yang, Kexiong Chen, *Study on statistics iterative thresholding segmentation based on aviation image*, **2** (2007), 187 –188.
- [26] Eduardo Rocha Costa Hemerson Pistori, Jeferson Pistori, *Hough-circles: Um módulo de detecção de circunferências para o imagej*.
- [27] Paul Hough, *Method and means for recognizing complex patterns*, United States Patent Office 3,069,654 (1960).
- [28] K.W.C. Foong P.S. Goh W.L. Nowinski H.P. Ng, S.H. Ong, *Medical image segmentation using k-means clustering and improved watershed algorithm*, (2006), 61–65.
- [29] G.Musiol H.Muehlig I.N.Bronshtein, K.A.Semendyayev, *Handbook of mathematics*, 5th ed., Springer-Verlag, 2007.

- [30] J. Kittler J. Illingworth, *A survey of the hough transform.*, Computer Vision, Graphics and Image Processing 44 (1988), 87–116.
- [31] Hong Zhang Jichuan Shi, *Adaptive local threshold with shape information and its application to object segmentation*, (2009), 1123–1128.
- [32] Dimitis G. Manolakis John G. Proakis, *Digital signal processing, principles, algorithms and applications*, 3th ed., Prentice-Hall InternaTional, 1996.
- [33] J. F. Reid K. Liao, Marvin R. Paulsen.
- [34] F. Tomita K. Sugimoto, *Boundary segmentation by detection of corner, inflection and transition points*, (1994), 13 –17.
- [35] H.G. Kaganami and Zou BeiJi, *Region-based segmentation versus edge detection*, (2009), 1217 –1221.
- [36] D.J. Kerbyson and T.J. Atherton, *Circle detection using hough transform filters*, (1995), 370–374.
- [37] Daniel T. Larose, *Discovering knowledge in data, an introduction to data mining*, Wiley, 2005.
- [38] K. I. Laws, *Rapid texture identification*, Image processing for missile guidance; Proceedings of the Seminar, San Diego, CA, July 29-August 1 (1980).
- [39] Isabel Lourtie, *Sinais e sistemas*, 2nd ed., Escolar Editora, 2007.
- [40] Bruce Batchelor Mark Graves, *Machine vision for the inspection of natural products*, Springer, 2004.
- [41] Alberto S. Aguado Mark S. Nixon, *Feature extraction and image processing*, Newnes, 2002.
- [42] F. Mário Martins, *Programação orientada aos objectos em java 2*, FCA, 2000.
- [43] José Maia Neves Miguel Rocha, Paulo Cortez, *Análise inteligente de dados, algoritmos e implementação em java*, FCA, 2008.
- [44] Chi-Man Pun Moon-Chuen Lee, *Classifying rotated textures using wavelet packet signatures*, (2000), no. 6, 2215 –2218 vol.4.
- [45] R.L. Baker N. Kanopoulos, N. Vasanthavada, *Design of an image edge detection filter using the sobel operator*, IEEE Journal of Solid-State Circuits 23 (1988), 358 – 367.

- [46] Thuy Tuong Nguyen, Xuan Dai Pham, Dongkyun Kim, and Jae Wook Jeon, *A test framework for the accuracy of line detection by hough transforms*, (2008), 1528 –1533.
- [47] Carla Oliveira, *Os queijos tradicionais do alentejo*, ASA/Porto, 1994.
- [48] Otsu, *A threshold selection method from gray-level histograms*, IEEE Transactions on Systems, Man and Cybernetics **9** (1979), no. 1, 62 –66.
- [49] Chih-Jen Lin Pai-Hsuen Chen, Rong-En Fan, *Training support vector machines via smo-type decomposition methods*.
- [50] John C. Platt, *Sequential minimal optimization: A fast algorithm for training support vector machines*, Technical Report MSR-TR-98-14 (1998).
- [51] William Prat, *Digital image processing*, 4th ed., Wiley, 2007.
- [52] Yong-Mei Cheng Quan-Xue Gao Qi-Chuan Tjan, Quan Pan, *Fast algorithm and application of hough transform in iris segmentation*, Third International Conference on Machine Learning and Cybernetics (2004).
- [53] Richard E. Woods Rafael C. Gonzalez, *Digital image processing*, Prentice Hall, 1992.
- [54] Its’Hak Dinstein Robert M. Haralick, K. Shanmugam, *Textural features for image classification*, IEEE Transactions on Systems, Man and Cybernetics **3** (1973), no. 6, 610 –621.
- [55] Azriel Rosenfeld, *The max roberts operator is a hueckel-type edge detector*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-3** (1981), 101 – 103.
- [56] Warren S. Sarle, *Stopped training and other remedies for overfitting*, (1995), 352–360.
- [57] Robert E. Schapire, *The strength of weak learnability*, Machine Learning **5** (1990), 197–227.
- [58] C.P. Schofield, *Evaluation of image analysis as a means of estimating the weight of pigs*, Journal of Agricultural Engineering Research **47** (1990), 287 – 296.
- [59] B. A. Sehnoi, *Introduction to digital signal processing and filter design*, Wiley, 2006.
- [60] Haijian Shi, *Best-first decision tree learning*, Master’s thesis, The University of Waikato, 2006.
- [61] Guan Yong Shi Na, Liu Xumin, *Research on k-means clustering algorithm: An improved k-means clustering algorithm*, (2010), 63–67.

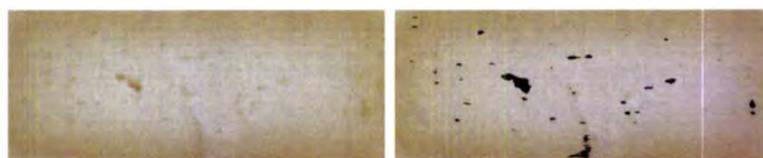
- [62] Lihua Wu Hongjian Wei Sixuan Zhao Shuqian He, Jiangqun Ni, *Image threshold segmentation method with 2-d histogram based on multi-resolution analysis*, (2009), 753-757.
- [63] Steven W. Smith, *The scientist and engineer's guide to digital signal processing*, 2nd ed., California Technical Publishing, 1999.
- [64] Jae Wook Jeon Thuy Tuong Nguyen, Xuan Dai Pham, *An improvement of the standard hough transform to detect line segments*, IEEE International Conference on Industrial Technology, 2008, pp. 1 – 6.
- [65] Qi-Chuan Tian, Quan Pan, Yong-Mei Cheng, and Quan-Xue Gao, *Fast algorithm and application of hough transform in iris segmentation*, **7** (2004), 3977 – 3980.
- [66] Cláudio Vieira Vagner Morais, *Matlab 6 e 7 - curso completo*, FCA, 2006.
- [67] Saeed V. Vaseghi, *Advanced digital signal processing and noise reduction*, John Wiley Sons Ltd, 2000.
- [68] L. Vibha, G.M. Harshavardhan, K. Pranaw, P.D. Shenoy, K.R. Venugopal, and L.M. Patnaik, *Statistical classification of mammograms using random forest classifier*, (2006), 178 –183.
- [69] Zhou Shisheng Wang Dong, *Color image recognition method based on the prewitt operator*, International Conference on Computer Science and Software Engineering, vol. 6, 2008, pp. 170 – 173.
- [70] GE Zhihui WEI Yanyan, LI Taoshen, *Combining distributed classifiers by stacking*, Third International Conference on Genetic and Evolutionary Computing (2009).
- [71] Mark Burge Wilhelm Burger, *Digital image processing an algorithmic introduction using java*, Springer, 2007.
- [72] Khin Hay Mar Saw Hla Win Thanda Aung, *Random forest classifier for multi-category classification of web pages*, (2009), 372 –376.
- [73] Ian H. Witten and Eibe Frank, *Data mining, practical machine learning tools and techniques*, Elsevier, 2005.
- [74] Mingyue Ding Xiaowei Fu, *A multi-threshold image segmentation method with adaptive fuzzy entropy*, (2009), 171 –174.
- [75] N.V. Boulgouris Xiayi Huang, *Robust object segmentation using adaptive thresholding*, **1** (2007).

- [76] K. S. Chen Y. C. Tzeng, S. H. Chiu, *Improvement of remote sensing image classification accuracy by using a multiple classifiers system with modified bagging and boosting algorithms.*
- [77] Zekui Zhou Yang Cui, *Application of pattern recognition on classifying texture images,* (2004).

Apêndice A

Apêndice ao capítulo: Segmentação

A.1 Resultados do algoritmo de segmentação



(a) Imagem original 6954a (b) Resultado da Segmentação

Figura A.1: Segmentação da imagem 6954a.jpg (1).



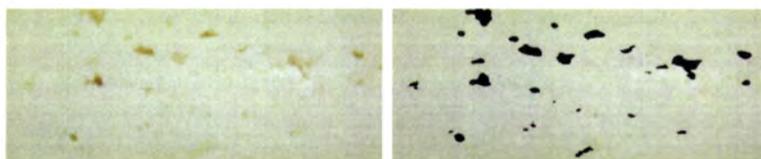
(a) Imagem original 2521d (b) Resultado da Segmentação

Figura A.2: Segmentação da imagem 2521d.jpg (2).



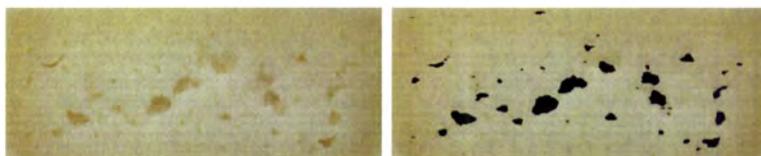
(a) Imagem original 3124a (b) Resultado da Segmentação

Figura A.3: Segmentação da imagem 3124a.jpg (3).



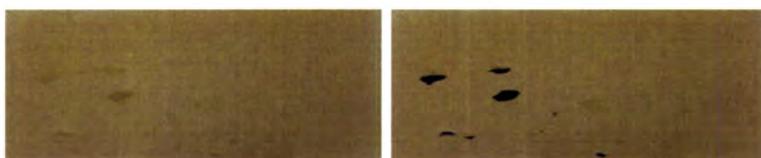
(a) Imagem original 6797a (b) Resultado da Segmentação

Figura A.4: Segmentação da imagem 6797a.jpg (4).



(a) Imagem original 2920a (b) Resultado da Segmentação

Figura A.5: Segmentação da imagem 2920a.jpg (5).



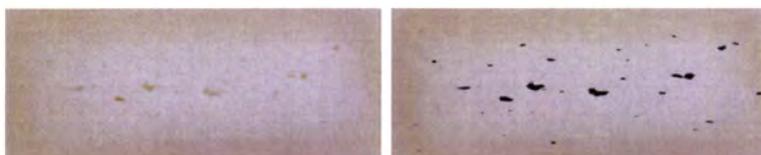
(a) Imagem original 2876a (b) Resultado da Segmentação

Figura A.6: Segmentação da imagem 2876a.jpg (6).



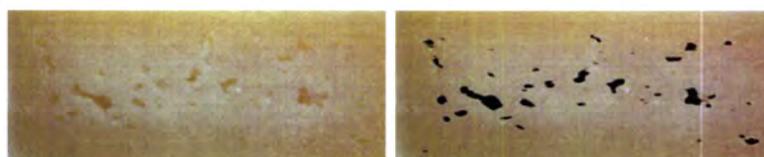
(a) Imagem original 9159a (b) Resultado da Segmentação

Figura A.7: Segmentação da imagem 9159a.jpg (7).



(a) Imagem original 4936a (b) Resultado da Segmentação

Figura A.8: Segmentação da imagem 4936a.jpg (8).



(a) Imagem original 3124b (b) Resultado da Segmentação

Figura A.9: Segmentação da imagem 3124b.jpg (9).



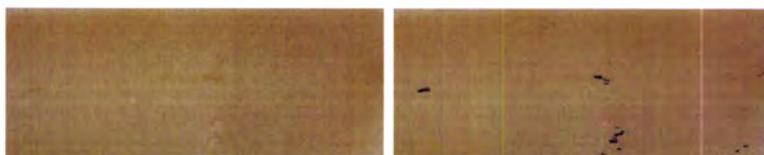
(a) Imagem original 4190a (b) Resultado da Segmentação

Figura A.10: Segmentação da imagem 4190a.jpg (10).



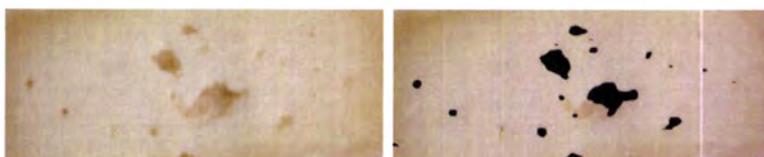
(a) Imagem original 6797c (b) Resultado da Segmentação

Figura A.11: Segmentação da imagem 6797c.jpg (11).



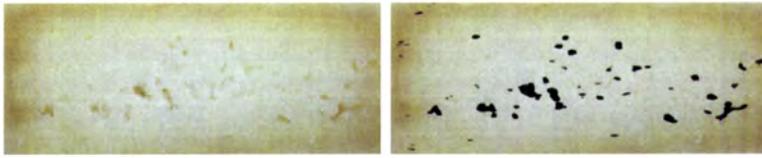
(a) Imagem original 5946a (b) Resultado da Segmentação

Figura A.12: Segmentação da imagem 5946a.jpg (12).



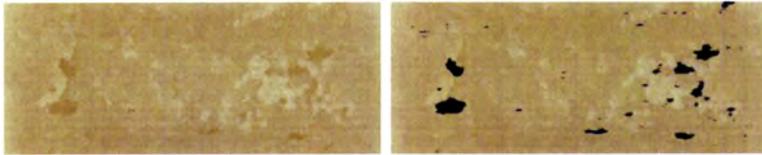
(a) Imagem original 9874a (b) Resultado da Segmentação

Figura A.13: Segmentação da imagem 9874a.jpg (13).



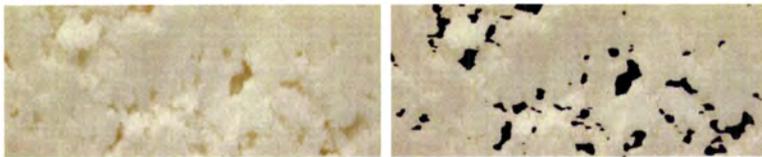
(a) Imagem original 8895b (b) Resultado da Segmentação

Figura A.14: Segmentação da imagem 8895b.jpg (14).



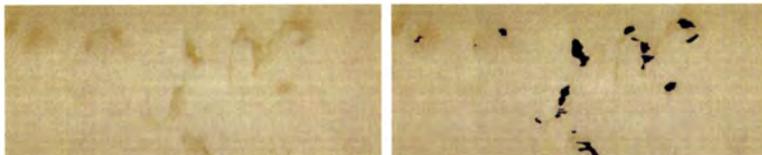
(a) Imagem original 8373a (b) Resultado da Segmentação

Figura A.15: Segmentação da imagem 8373a.jpg (15).



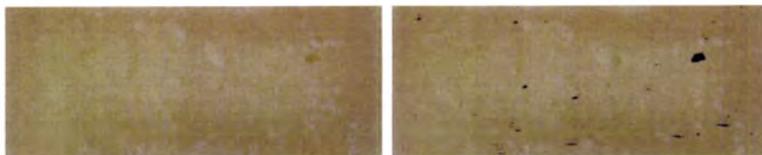
(a) Imagem original 7843b (b) Resultado da Segmentação

Figura A.16: Segmentação da imagem 7843b.jpg (16).



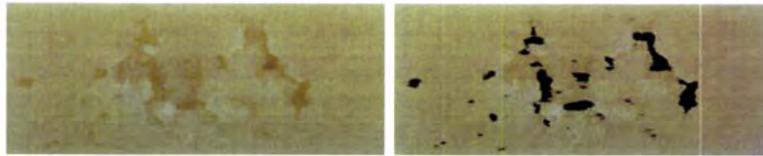
(a) Imagem original 5437a (b) Resultado da Segmentação

Figura A.17: Segmentação da imagem 5437a.jpg (17).



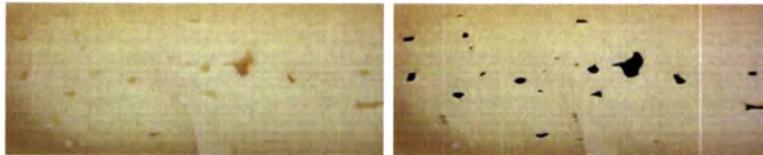
(a) Imagem original 4856a (b) Resultado da Segmentação

Figura A.18: Segmentação da imagem 4856a.jpg (18).



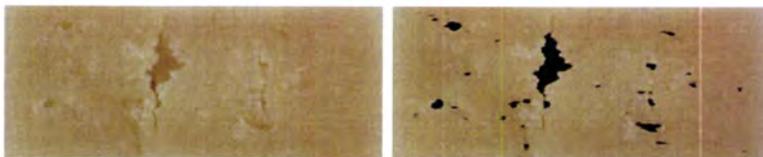
(a) Imagem original 8352b (b) Resultado da Segmentação

Figura A.19: Segmentação da imagem 8352b.jpg (19).



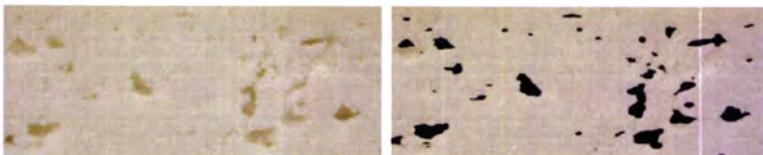
(a) Imagem original 7091a (b) Resultado da Segmentação

Figura A.20: Segmentação da imagem 7091a.jpg (20).



(a) Imagem original 2521c (b) Resultado da Segmentação

Figura A.21: Segmentação da imagem 2521c.jpg (21).



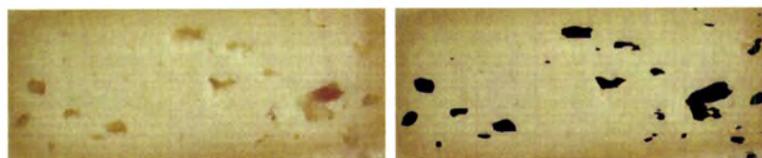
(a) Imagem original 7843a (b) Resultado da Segmentação

Figura A.22: Segmentação da imagem 7843a.jpg (22).



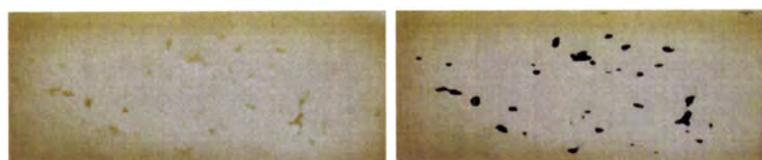
(a) Imagem original 4856b (b) Resultado da Segmentação

Figura A.23: Segmentação da imagem 4856b.jpg (23).



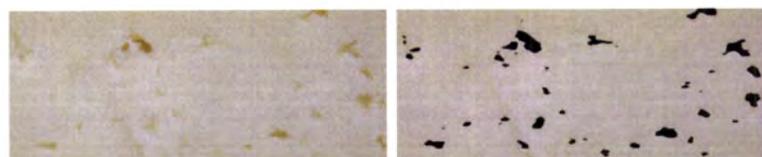
(a) Imagem original 2103a (b) Resultado da Segmentação

Figura A.24: Segmentação da imagem 2103a.jpg (24).



(a) Imagem original 8895a (b) Resultado da Segmentação

Figura A.25: Segmentação da imagem 8895a.jpg (25).



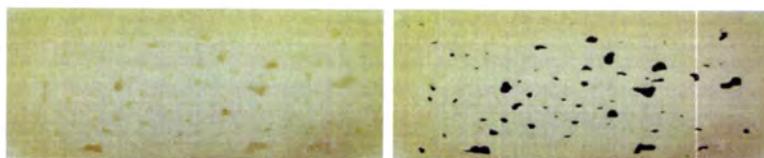
(a) Imagem original 6797b (b) Resultado da Segmentação

Figura A.26: Segmentação da imagem 6797b.jpg (26).



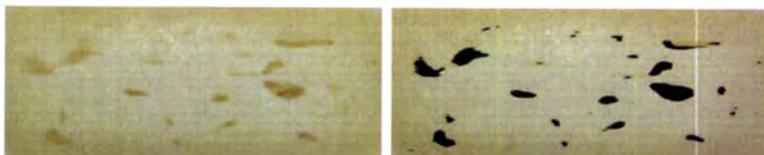
(a) Imagem original 7091b (b) Resultado da Segmentação

Figura A.27: Segmentação da imagem 7091b.jpg (27).



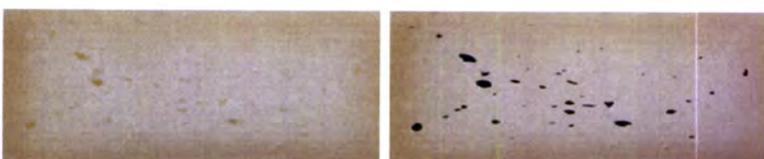
(a) Imagem original 2920b (b) Resultado da Segmentação

Figura A.28: Segmentação da imagem 2920b.jpg (28).



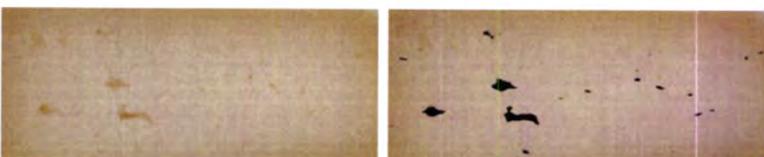
(a) Imagem original 9159c (b) Resultado da Segmentação

Figura A.29: Segmentação da imagem 9159c.jpg (29).



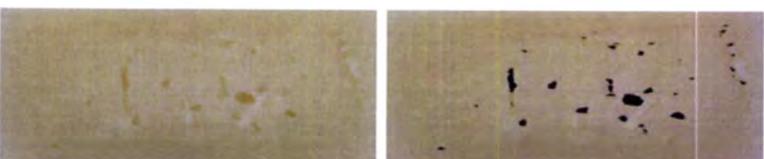
(a) Imagem original 4936b (b) Resultado da Segmentação

Figura A.30: Segmentação da imagem 4936b.jpg (30).



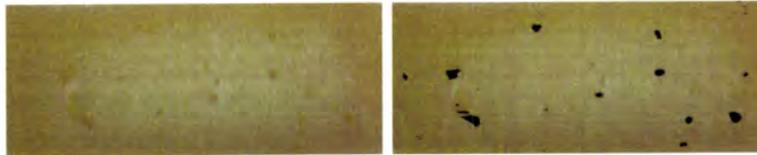
(a) Imagem original 9159b (b) Resultado da Segmentação

Figura A.31: Segmentação da imagem 9159b.jpg (31).



(a) Imagem original 8352a (b) Resultado da Segmentação

Figura A.32: Segmentação da imagem 8352a.jpg (32).



(a) Imagem original 7540a

(b) Resultado da Segmentação

Figura A.33: Segmentação da imagem 7540a.jpg (33).



(a) Imagem original 2521e

(b) Resultado da Segmentação

Figura A.34: Segmentação da imagem 2521e.jpg (34).

A.2 Valores da segmentação para cada filtro

Na tabela A.1 são indicados os valores registados para os diversos filtros aplicados a cada imagem, aquando do processo de segmentação. São igualmente indicados os valores para as 4 primeiras iterações do filtro não linear (NL).

Tabela A.1: Resultado do algoritmo de segmentação nas diversas iterações.

Número	Foto	Colunas	Linhas	Filtro Linear	1ª It. NL	2ª It. NL	3ª It. NL	4ª It. NL
1	6954a	779	283	2204	2693	2785	2806	2808
2	2521d	1171	553	7027	8440	8672	8749	8770
3	3124a	646	292	2561	3127	3221	3239	3247
4	6797a	1031	380	9581	12106	12656	12820	12868
5	2920a	1652	423	21529	26767	27849	28091	28140
6	2876a	1687	681	7288	9626	10391	10655	10733
7	9159a	1231	595	14963	19716	21265	21876	22137
8	4936a	1598	674	7325	8977	9255	9297	9307
9	3124b	2048	761	33635	41713	43450	43849	43951
10	4190a	1358	480	6031	7334	7540	7577	7589
11	6797c	915	348	18250	23645	24972	25246	25318
12	5946a	573	266	395	481	492	493	493
13	9874a	1181	340	9911	13334	14420	14785	14906
14	8895b	1626	472	15720	19688	20586	20858	20970
15	8373a	899	327	5555	7251	7725	7897	7958
16	7843b	1174	322	20218	25828	27354	27848	28056
17	5437a	1102	295	4580	5877	6249	6349	6388
18	4856a	1272	440	2026	2324	2383	2397	2399
19	8352b	1034	487	15181	18512	20681	21056	21203
20	7091a	2048	835	22962	27963	28722	28894	28956
21	2521c	1370	489	15530	18967	19561	19727	19820
22	7843a	972	340	15314	19201	20005	20168	20186
23	4856b	1208	506	7378	9546	10052	10179	10209
24	2103a	993	315	12592	15977	16985	17392	17556
25	8895a	1564	455	10970	13289	13683	13756	13779
26	6797b	1122	352	11300	13885	14355	14433	14452
27	7091b	2048	976	34090	42089	43580	44046	44282
28	2920b	1432	595	18759	22948	23767	23966	24019
29	9159c	1073	541	17531	23107	24722	25302	25533
30	4936b	1772	707	12395	14759	15103	15155	15174
31	9159b	1409	536	6920	9086	9593	9702	9725
32	8352a	1125	543	7276	8722	8943	8980	8989
33	7540a	1544	438	6007	7322	7580	7650	7692
34	2521e	1689	659	16602	20711	21687	21932	22008

A.3 Limiares determinados pelo método de Otsu

Na tabela A.2, são apresentados os valores dos limiares calculados pelo método do Otsu. A coluna Limiares Parciais, corresponde aos limiares calculados para cada uma das 16 imagens, resultantes da divisão da imagem original em 16 regiões de iguais dimensões. A coluna Limiar Total apresenta o limiar utilizado pelo método de Otsu para realizar a segmentação das imagens da coluna Foto.

Tabela A.2: Valores dos limiares parciais e limiar total pelo método de Otsu.

Foto	Limiares Parciais				Limiar Total
2103a	125	140	144	133	135
	129	153	140	141	
	125	137	136	112	
	126	133	134	131	
2876a	102	99	91	89	88
	93	96	90	86	
	95	82	86	81	
	90	91	79	80	
6797c	145	130	136	140	133
	142	147	142	137	
	127	137	140	126	
	148	131	131	148	

Apêndice B

Apêndice ao capítulo: Detecção de Texturas

B.1 Resultados do algoritmo de detecção de texturas

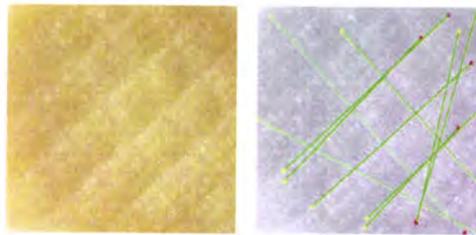


Figura B.1: Resultado da aplicação da transformada de Hough, imagem 1.

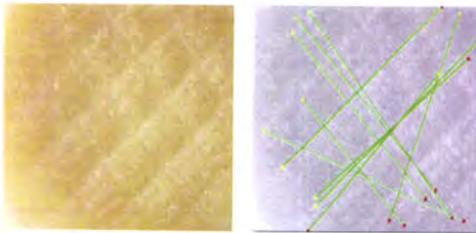


Figura B.2: Resultado da aplicação da transformada de Hough, imagem 2.

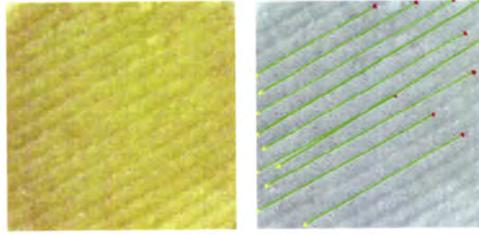


Figura B.3: Resultado da aplicação da transformada de Hough, imagem 3.

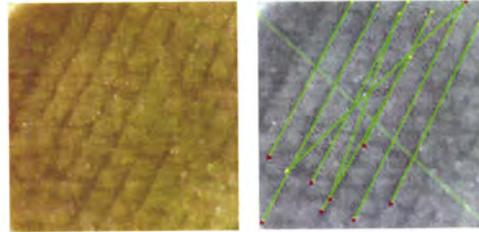


Figura B.4: Resultado da aplicação da transformada de Hough, imagem 4.

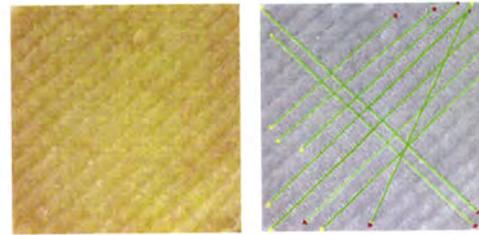


Figura B.5: Resultado da aplicação da transformada de Hough, imagem 5.

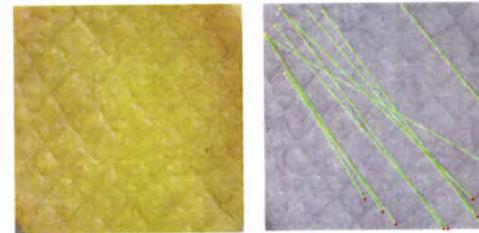


Figura B.6: Resultado da aplicação da transformada de Hough, imagem 6.

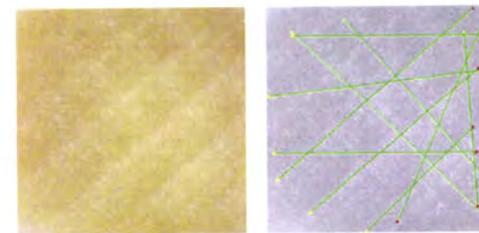


Figura B.7: Resultado da aplicação da transformada de Hough, imagem 7.

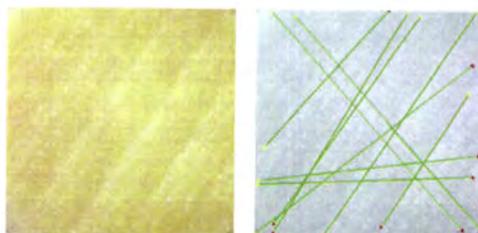


Figura B.8: Resultado da aplicação da transformada de Hough, imagem 8.

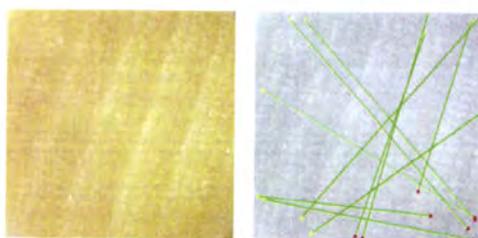


Figura B.9: Resultado da aplicação da transformada de Hough, imagem 9.

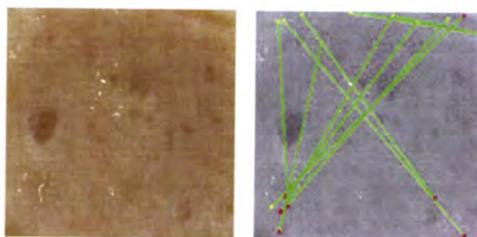


Figura B.10: Resultado da aplicação da transformada de Hough, imagem 10.

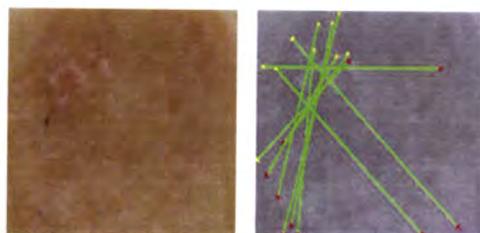


Figura B.11: Resultado da aplicação da transformada de Hough, imagem 11.

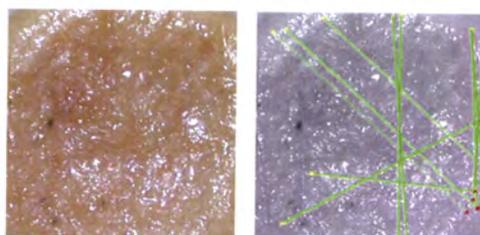


Figura B.12: Resultado da aplicação da transformada de Hough, imagem 12.

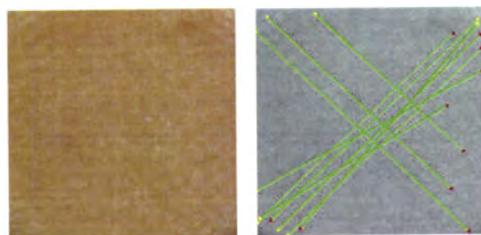


Figura B.13: Resultado da aplicação da transformada de Hough, imagem 13.

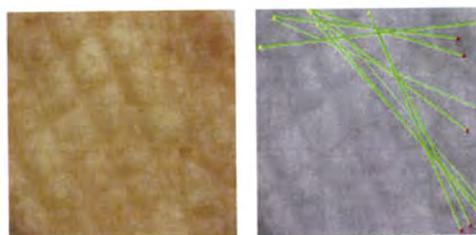


Figura B.14: Resultado da aplicação da transformada de Hough, imagem 14.

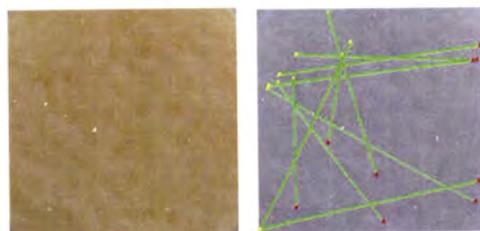


Figura B.15: Resultado da aplicação da transformada de Hough, imagem 15.

B.2 Valores da detecção de textura

A tabela B.1 apresenta a pontuação de cada queijo (secção B.1) relativamente à presença de textura, segundo a metodologia descrita em 6.4.

Tabela B.1: Resultados da aplicação da transformada de Hough a diversas imagens.

Imagem	Avaliação
1	22.8740
2	26.8866
3	44.6143
4	29.7972
5	36.0494
6	31.5718
7	15.2172
8	20.4280
9	14.2304
10	15.2772
11	15.3209
12	16.7708
13	33.2805
14	10.2446
15	13.1425

Apêndice C

Apêndice ao capítulo: Recolha de Atributos

C.1 Lista de atributos de cada imagem

Número de Olhos - Número de regiões classificadas como olhos do queijo, pelo algoritmo de segmentação, e contadas pelo algoritmo Analyse Particles, excepto olhos com área inferior a 5 pixels.

Número Olhos (total) - Contagem de todas as regiões classificadas como olhos do queijo pelo algoritmo Analyse Particles.

Área Total - Área total classificada como olhos, em pixels.

Área Média - Área média dos olhos, correspondendo ao quociente entre o número de olhos, excepto olhos com área inferior a 5 pixels, e a área total ocupada pelos olhos, em pixels.

Circularidade - Cálculo da circularidade média dos olhos, pela equação 7.1.

Fracção da Área - Percentagem de área de olhos relativamente á área total do queijo.

Pixels Menores 40 Horizontal - Sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das linhas da imagem;

Pixels Maiores 40 Horizontal - Sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das linhas da imagem;

Olhos Menores 40 Horizontal - Número de sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das linhas da imagem;

Olhos Maiores 40 Horizontal - Número de sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das linhas da imagem;

Pixels Menores 35 Horizontal - Sequências de pixels consecutivos inferiores a 35 pixels, contados em cada uma das linhas da imagem;

Pixels Maiores 35 Horizontal - Sequências de pixels consecutivos superiores ou iguais a 35 pixels, contados em cada uma das linhas da imagem;

Olhos Menores 23 Horizontal - Número de sequências de pixels consecutivos inferiores a 23 pixels, contados em cada uma das linhas da imagem;

Olhos Maiores 23 Horizontal - Número de sequências de pixels consecutivos superiores ou iguais a 23 pixels, contados em cada uma das linhas da imagem;

Pixels Menores 40 Vertical - Sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das colunas da imagem;

Pixels Maiores 40 Vertical - Sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das colunas da imagem;

Olhos Menores 40 Vertical - Número de sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das colunas da imagem;

Olhos Maiores 40 Vertical - Número de sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das linhas da imagem;

Pixels Menores 40 Diagonal - Sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das diagonais da imagem;

Pixels Maiores 40 Diagonal - Sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das diagonais da imagem;

Olhos Menores 40 Diagonal - Número de sequências de pixels consecutivos inferiores a 40 pixels, contados em cada uma das diagonais da imagem;

Olhos Maiores 40 Diagonal - Número de sequências de pixels consecutivos superiores ou iguais a 40 pixels, contados em cada uma das diagonais da imagem;

Média R Centro - Média da componente *Red*, do espaço de cores RGB, no centro do queijo.

Média G Centro - Média da componente *Green*, do espaço de cores RGB, no centro do queijo.

Média B Centro - Média da componente *Blue*, do espaço de cores RGB, no centro do queijo.

- Média Gray Centro** - Média na escala de cinzentos, para o espaço de cores RGB, no centro do queijo.
- Média R Crosta** - Média da componente *Red*, do espaço de cores RGB, na crosta do queijo.
- Média G Crosta** - Média da componente *Green*, do espaço de cores RGB, na crosta do queijo.
- Média B Crosta** - Média da componente *Blue*, do espaço de cores RGB, na crosta do queijo.
- Média Gray Crosta** - Média na escala de cinzentos, para o espaço de cores RGB, na crosta do queijo.
- Máximo Centro Crosta R** - Diferença entre a média da componente *Red* do centro e a componente *Red* da crosta do queijo.
- Máximo Centro Crosta G** - Diferença entre a média da componente *Green* do centro e a componente *Green* da crosta do queijo.
- Máximo Centro Crosta B** - Diferença entre a média da componente *Blue* do centro e a componente *Blue* da crosta do queijo.
- Máximo Centro Crosta Gray** - Diferença entre a média na escala de cinzentos do centro e a média na escala de cinzentos da crosta do queijo.
- Média H Centro** - Média da componente *Hue*, do espaço de cores HSV, no centro do queijo.
- Média S Centro** - Média da componente *Saturation*, do espaço de cores HSV, no centro do queijo.
- Média V Centro** - Média da componente *Value*, do espaço de cores HSV, no centro do queijo.
- Média H Crosta** - Média da componente *Hue*, do espaço de cores HSV, na crosta do queijo.
- Média S Crosta** - Média da componente *Saturation*, do espaço de cores HSV, na crosta do queijo.
- Média V Crosta** - Média da componente *Value*, do espaço de cores HSV, na crosta do queijo.
- Diferença H Centro Crosta** - Diferença entre a média da componente *Hue* do centro e a componente *Hue* da crosta do queijo.

Diferença S Centro Crosta - Diferença entre a média da componente *Saturation* do centro e a componente *Saturation* da crosta do queijo.

Diferença V Centro Crosta - Diferença entre a média da componente *Value* do centro e a componente *Value* da crosta do queijo.

Média R Canto Superior Esquerdo - Média da componente *Red*, do espaço de cores RGB, no canto superior esquerdo do queijo.

Média G Canto Superior Esquerdo - Média da componente *Green*, do espaço de cores RGB, no canto superior esquerdo do queijo.

Média B Canto Superior Esquerdo - Média da componente *Blue*, do espaço de cores RGB, no canto superior esquerdo do queijo.

Média Gray Canto Superior Esquerdo - Média na escala de cinzentos para o espaço de cores RGB, no canto superior esquerdo do queijo.

Média R Canto Superior Direito - Média da componente *Red*, do espaço de cores RGB, no canto superior direito do queijo.

Média G Canto Superior Direito - Média da componente *Green*, do espaço de cores RGB, no canto superior direito do queijo.

Média B Canto Superior Direito - Média da componente *Blue*, do espaço de cores RGB, no canto superior direito do queijo.

Média Gray Canto Superior Direito - Média na escala de cinzentos para o espaço de cores RGB, no canto superior direito do queijo.

Média R Canto Inferior Esquerdo - Média da componente *Red*, do espaço de cores RGB, no canto inferior esquerdo do queijo.

Média G Canto Inferior Esquerdo - Média da componente *Green*, do espaço de cores RGB, no canto inferior esquerdo do queijo.

Média B Canto Inferior Esquerdo - Média da componente *Blue*, do espaço de cores RGB, no canto inferior esquerdo do queijo.

Média Gray Canto Inferior Esquerdo - Média na escala de cinzentos para o espaço de cores RGB, no canto inferior esquerdo do queijo.

Média R Canto Inferior Direito - Média da componente *Red*, do espaço de cores RGB, no canto inferior direito do queijo.

- Média G Canto Inferior Direito** - Média da componente *Green*, do espaço de cores RGB, no canto inferior direito do queijo.
- Média B Canto Inferior Direito** - Média da componente *Blue*, do espaço de cores RGB, no canto inferior direito do queijo.
- Média Gray Canto Inferior Direito** - Média na escala de cinzentos para o espaço de cores RGB, no canto inferior direito do queijo.
- Máximo Centro Cantos R** - Valor máximo da diferença entre a média da componente *Red* do centro do queijo, por comparação com cada um dos quarto cantos do queijo.
- Máximo Centro Cantos G** - Valor máximo da diferença entre a média da componente *Green* do centro do queijo, por comparação com cada um dos quarto cantos do queijo.
- Máximo Centro Cantos B** - Valor máximo da diferença entre a média da componente *Blue* do centro do queijo, por comparação com cada um dos quarto cantos do queijo.
- Máximo Centro Cantos Gray** - Valor máximo da diferença entre a média na escala de cinzentos do centro do queijo, por comparação com cada um dos quarto cantos do queijo.
- Máximo Centro Cantos Todos R** - Valor máximo da diferença entre a média da componente *Red* do centro do queijo, por comparação com cada um dos quarto cantos do queijo e dos cantos entre si.
- Máximo Centro Cantos Todos G** - Valor máximo da diferença entre a média da componente *Green* do centro do queijo, por comparação com cada um dos quarto cantos do queijo e dos cantos entre si.
- Máximo Centro Cantos Todos B** - Valor máximo da diferença entre a média da componente *Blue* do centro do queijo, por comparação com cada um dos quarto cantos do queijo e dos cantos entre si.
- Máximo Centro Cantos Todos Gray** - Valor máximo da diferença entre na escala de cinzentos do centro do queijo, por comparação com cada um dos quarto cantos do queijo e dos cantos entre si.
- Média R Fundo** - Média da componente *Red*, do espaço de cores RGB, da imagem resultante da subtração do resultado da segmentação à imagem original.
- Média G Fundo** - Média da componente *Green*, do espaço de cores RGB, da imagem resultante da subtração do resultado da segmentação à imagem original.

Média B Fundo - Média da componente *Blue*, do espaço de cores RGB, da imagem resultante da subtração do resultado da segmentação à imagem original.

Média H Fundo - Média da componente *Hue*, do espaço de cores HSV, da imagem resultante da subtração do resultado da segmentação à imagem original.

Média S Fundo - Média da componente *Saturation*, do espaço de cores HSV, da imagem resultante da subtração do resultado da segmentação à imagem original.

Média V Fundo - Média da componente *Value*, do espaço de cores HSV, da imagem resultante da subtração do resultado da segmentação à imagem original.

C.2 Características dos olhos pelo algoritmo *Analyze Particles*

Na tabela C.1 são apresentados os valores das características dos olhos obtidas pela aplicação do *plugin Analyze Particles*, às imagens binárias resultantes do algoritmo de segmentação descrito no capítulo 5.

C.2. CARACTERÍSTICAS DOS OLHOS PELO ALGORITMO ANALYZE PARTICLES121

Tabela C.1: Resultado da extracção das características dos olhos pelo *plugin Analyze Particles*.

Foto	Número de Olhos	Tamanho Médio	Fracção Área	Circularidade	Área Total
2103a	39	446.769	5.570	0.162	17446
2521c	39	501.128	2.917	0.126	19573
2521d	41	211.122	1.338	0.139	8692
2521e	58	373.034	1.944	0.125	21687
2876a	17	626.235	0.927	0.138	10668
2920a	45	618.622	3.984	0.173	27849
2920b	72	329.680	2.786	0.188	23767
3124a	36	88.167	1.683	0.226	3221
3124b	61	712.115	2.787	0.142	43457
4190a	33	227.848	1.153	0.196	7546
4856a	45	52.289	0.420	0.159	2431
4856b	48	211.729	1.662	0.136	10220
4936a	39	236.334	0.856	0.152	9257
4936b	52	290.115	1.204	0.193	15103
5437a	21	302.190	1.952	0.151	6349
5946a	21	22.714	0.313	0.171	506
6797a	31	407.742	3.226	0.181	12656
6797b	45	318.733	3.632	0.163	14358
6797c	26	971	7.928	0.139	25246
6954a	40	68.975	1.251	0.152	2788
7091a	43	667.325	1.678	0.174	28722
7091b	20	2197	2.180	0.190	43582
7540a	33	228.667	1.116	0.173	7594
7843a	43	465.186	6.053	0.161	20010
7843b	64	434.969	7.364	0.141	27848
8352a	44	203.023	1.462	0.205	8948
8352b	43	489.186	4.177	0.132	21056
8373a	50	157.180	2.673	0.170	7908
8895a	40	342.6	1.926	0.175	13722
8895b	64	321.437	2.680	0.156	20594
9159a	45	485.333	2.982	0.155	21876
9159b	27	368.815	1.255	0.134	10010
9159c	42	603.286	4.365	0.170	25349
9874a	22	671.636	3.680	0.176	14787

Apêndice D

Apêndice ao Capítulo: Aprendizagem e Classificação

D.1 Resumo das classificações do *nível-0*

Na tabela D.1, são apresentados os resultados dos diferentes classificadores, para cada um dos parâmetros em que assenta a classificação de um queijo.

D.2 Resumo das classificações do *nível-1*

Na tabela D.2, são apresentados os resultados da classificação final com o esquema de *Stacking*, para todas as imagens que integram o *corpus*, que foi utilizado para treino e teste do classificador.

Tabela D.1: Resumo do comportamento de cada classificador do *nível-0*.

Foto	Cor	SMO	Olhos	J48	Homogeneidade	BFTrec	Bons Maus	ADTree
2103a	Centro Escuro	✓	Muitos	✓	Não	✓	Bom	✓
2521c	Centro Escuro	✓	Poucos	Muitos	Sim	✓	Bom	✓
2521d	Centro Escuro	✓	Poucos	✓	Sim	✓	Bom	✓
2521e	Centro Escuro	✓	Médios	✓	Sim	✓	Bom	✓
2876a	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
2920a	Centro Escuro	✓	Muitos	✓	Sim	Não	Bom	✓
2920b	Centro Escuro	✓	Médios	✓	Não	✓	Bom	✓
3124a	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
3124b	Centro Escuro	✓	Médios	✓	Não	✓	Bom	✓
4190a	Centro Claro	CE	Poucos	✓	Não	✓	Bom	✓
4856a	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
4856b	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
4936a	Centro Claro	✓	Poucos	✓	Não	✓	Bom	Mau
4936b	Centro Claro	✓	Poucos	✓	Não	✓	Bom	✓
5437a	Centro Escuro	✓	Médios	✓	Não	Sim	Bom	✓
5946a	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
6797a	Branco	✓	Muitos	✓	Sim	✓	Mau	✓
6797b	Branco	✓	Muitos	✓	Sim	✓	Mau	✓
6797c	Branco	✓	Muitos	✓	Sim	✓	Mau	✓
6954a	Centro Claro	✓	Poucos	✓	Não	✓	Bom	✓
7091a	Centro Escuro	✓	Poucos	✓	Não	✓	Bom	✓
7091b	Branco	CC	Médios	✓	Não	✓	Mau	Bom
7540a	Escuro	✓	Poucos	✓	Sim	✓	Mau	✓
7843a	Branco	✓	Muitos	✓	Sim	✓	Mau	✓
7843b	Branco	✓	Muitos	✓	Sim	✓	Mau	✓
8352a	Escuro	✓	Poucos	✓	Sim	✓	Bom	✓
8352b	Escuro	✓	Muitos	✓	Sim	✓	Mau	✓
8373a	Centro Escuro	✓	Médios	✓	Sim	✓	Bom	✓
8895a	Centro Claro	✓	Poucos	Médios	Não	✓	Bom	✓
8895b	Centro Claro	✓	Médios	✓	Não	✓	Bom	✓
9159a	Centro Escuro	✓	Muitos	✓	Não	✓	Bom	✓
9159b	Centro Escuro	✓	Poucos	✓	Não	✓	Bom	✓
9159c	Centro Escuro	✓	Muitos	✓	Não	✓	Bom	✓
9874a	Centro Escuro	✓	Muitos	✓	Não	✓	Bom	✓

Nota: CE - Centro Escuro; CC - Centro Claro

Tabela D.2: Resultado das classificações do *nível-1*.

Foto	Classe	Cor	Olhos	Homogeneidade	Bons Maus	Stacking
2103a	Insuficiente	Centro Escuro	Muitos	Não	Bom	✓
2521c	Muito Bom	Centro Escuro	Muitos	Sim	Bom	✓
2521d	Bom	Centro Escuro	Poucos	Sim	Bom	✓
2521e	Muito Bom	Centro Escuro	Médios	Sim	Bom	✓
2876a	Mau	Escuro	Poucos	Sim	Mau	✓
2920a	Insuficiente	Centro Escuro	Muitos	Não	Bom	✓
2920b	Bom	Centro Escuro	Médios	Não	Bom	Muito Bom
3124a	Mau	Escuro	Poucos	Sim	Mau	✓
3124b	Muito Bom	Centro Escuro	Médios	Não	Bom	Bom
4190a	Suficiente	Centro Escuro	Poucos	Não	Bom	Bom
4856a	Mau	Escuro	Poucos	Sim	Mau	✓
4856b	Mau	Escuro	Poucos	Sim	Mau	✓
4936a	Suficiente	Centro Claro	Poucos	Não	Mau	✓
4936b	Suficiente	Centro Claro	Poucos	Não	Bom	✓
5437a	Insuficiente	Centro Escuro	Médios	Sim	Bom	Muito Bom
5946a	Mau	Escuro	Poucos	Sim	Mau	✓
6797a	Mau	Branco	Muitos	Sim	Mau	✓
6797b	Mau	Branco	Muitos	Sim	Mau	✓
6797c	Mau	Branco	Muitos	Sim	Mau	✓
6954a	Suficiente	Centro Claro	Poucos	Não	Bom	✓
7091a	Bom	Centro Escuro	Poucos	Não	Bom	✓
7091b	Mau	Centro Claro	Médios	Não	Bom	Suficiente
7540a	Mau	Escuro	Poucos	Sim	Mau	✓
7843a	Mau	Branco	Muitos	Sim	Mau	✓
7843b	Mau	Branco	Muitos	Sim	Mau	✓
8352a	Bom	Escuro	Poucos	Sim	Bom	✓
8352b	Mau	Escuro	Muitos	Sim	Mau	✓
8373a	Muito Bom	Centro Escuro	Médios	Sim	Bom	✓
8895a	Suficiente	Centro Claro	Médios	Não	Bom	✓
8895b	Suficiente	Centro Claro	Médios	Não	Bom	✓
9159a	Insuficiente	Centro Escuro	Muitos	Não	Bom	✓
9159b	Bom	Centro Escuro	Poucos	Não	Bom	✓
9159c	Insuficiente	Centro Escuro	Muitos	Não	Bom	✓
9874a	Insuficiente	Centro Escuro	Muitos	Não	Bom	✓

24