

André Teixeira Gomes

IDENTIFICAÇÃO E RECONHECIMENTO DE PÓLIPOS EM IMAGENS DE COLONOSCOPIAS.

Dissertação de Mestrado

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Setembro de 2014



UNIVERSIDADE DE COIMBRA



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Electrotécnica e de Computadores

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Título Identificação e Reconhecimento de Pólipos em Imagens de Colonoscopias

Candidato André Teixeira Gomes

Júri João Pedro de Almeida Barreto
Helder de Jesus Araújo
Jorge Nuno de Almeida e Sousa Almada Lobo

Data Setembro de 2014

Agradecimentos

A realização desta dissertação foi possível graças aos importantes apoios e incentivos que tive a felicidade de receber.

Ao Professor Doutor Helder de Jesus Araújo, Professor Catedrático do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, pela sua orientação, apoio e disponibilidade ao longo de toda a duração deste trabalho.

À minha família, namorada, amigos e colegas, que sempre me acompanharam e motivaram ao longo desta importante etapa.

Ao projecto FCT/PTDC/EIA-EIA/122454/2010, financiado pela Fundação para a Ciência e Tecnologia por meio do PIDDAC e FEDER através do programa operacional COMPETE.

Resumo

Esta dissertação apresenta um estudo comparativo entre os métodos *Speeded Up Robust Features* (SURF) e *Scale-Invariant Feature Transform* (SIFT), como métodos de obtenção de *features* locais, e entre um Máquina de Vetores de Suporte (SVM) e Árvores de Classificação, como método de classificação. Este estudo foi realizado no âmbito da identificação e reconhecimento de pólipos em imagens de colonoscopias. Foi aplicada uma seleção da Região de Interesse (ROI) baseada em morfologia matemática para eliminar artefactos intrínsecos do método de obtenção de imagens em colonoscopias. Complementarmente às *features* locais foram utilizadas *features* extraídas a partir da cor e da matriz de coocorrências em níveis de cinzento (GLCM). Foram realizadas várias experiências com a aplicação do método de Análise de Componentes Principais (PCA) e mostrou-se que utilizando os apenas quatro primeiros componentes principais da matriz de *features* se obtêm resultados suficientemente significativos para uma classificação robusta. Os métodos de classificação foram testados utilizando o método *leave-one-out* recorrendo a todo o conjunto de imagens disponíveis. Este trabalho conclui que o sistema com obtenção de *features* locais utilizando SURF é mais ágil computacionalmente e que as árvores de classificação produzem consistentemente melhores resultados no conjunto de imagens testado.

Palavras-chave: Colonoscopia, SURF, SIFT, PCA, SVM, Árvores Binárias.

Abstract

This paper presents a comparative study between Speeded Up Robust Features (SURF) and Scale-Invariant Feature Transform (SIFT) as methods for obtaining local features, and between Support Vectors Machine (SVM) and Classification Trees as classification methods. This study was conducted in the identification and recognition of polyps. A selection of the Region of Interest (ROI) based on mathematical morphology was applied to eliminate intrinsic artifacts originating from image acquisition system of the colonoscopy. In addition to the local features, features based on the image color and on the gray-level co-occurrence matrix (GLCM) were also used. Several experiments were made with the Principal Components Analysis (PCA) method and it was shown that using only the first four principal components of the feature matrix was enough for a robust classification. The classification methods were tested using the leave-one-out method through the full set of images available. This paper concludes that one system based on SURF is computationally faster and that classification trees produce consistently better results in the set of images tested.

Keywords: Colonoscopy, SURF, SIFT, PCA, SVM, Classification Trees.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	iv
1 Introdução	1
1.1 Enquadramento	1
1.2 Objetivos	2
1.3 Estrutura da Dissertação	3
2 Conceitos Fundamentais	5
2.1 Aquisição e Tratamento de Imagens	5
2.1.1 Colonoscopia	5
2.1.2 Pré-Processamento	5
2.2 Extração de Features	7
2.2.1 Features Baseadas na Cor	7
2.2.2 Features Locais	7
2.2.3 Features de Textura	8
2.3 Seleção estatística de <i>features</i>	11
2.4 Classificação	14

3	Trabalho Realizado	15
3.1	Imagens e Estrutura do Sistema	15
3.2	Pré-Processamento	16
3.3	Extração de Características	17
3.3.1	Características Baseadas na Cor	17
3.3.2	Features Locais	18
3.3.3	<i>Features</i> de Textura	20
3.4	Seleção estatística de <i>features</i>	21
3.5	Classificação	22
3.5.1	Aprendizagem	23
3.5.2	Extração de novas <i>features</i>	24
3.5.3	Classificação	25
3.6	Resumo	25
4	Resultados	28
5	Conclusões	33

Lista de Figuras

3.1	Sequência de passos da definição da região de interesse.	17
4.1	Curva <i>Precision vs Recall</i> para o sistema com os métodos SURF e Árvores Binárias.	32

Lista de Tabelas

2.1	As três características extraídas com base na cor das imagens. R, G e B são matrizes do tamanho da imagem em pixels que indicam qual o valor de cada componente (vermelho, verde e azul) para esse pixel.	7
3.1	Os 6 valores próprios com maior magnitude obtidos aquando da aplicação do PCA a um conjunto de dados com 201 observações e 665 variáveis.	22
3.2	Significado das três etiquetas usadas para identificação das imagens.	24
4.1	Tabela dos resultados de classificação em três categorias. VB, VN e VM são as taxas de classificação correta dos casos benignos, normais e malignos, respetivamente.	29
4.2	Matriz de confusão para SURF/Árvores Binárias.	29
4.3	Matriz de confusão para SURF/SVM.	30
4.4	Matriz de confusão para SIFT/Árvores Binárias.	30
4.5	Matriz de confusão para SIFT/SVM.	30
4.6	Taxas de falhas num classificação binária. FP e FN representam a percentagem de falsos positivos e de falsos negativos, respetivamente.	31

4.7	Precisão e <i>Recall</i> obtidos utilizando aquelas que provaram ser as melhores condições do sistema. O modelo baseado em árvores binárias é alguns pontos percentuais superior ao baseado em SVM em todas as métricas testadas e para ambos os algoritmos de obtenção de <i>features</i> locais.	31
-----	--	----

Capítulo 1

Introdução

1.1 Enquadramento

O cancro colo-retal é a terceira principal causa de morte por cancro em todo o mundo. Em Portugal é a segunda forma de cancro mais frequente e estima-se que, todos os anos, morram cerca de 2200 pessoas devido a esta doença oncológica [1]. Os principais sintomas incluem a alteração dos hábitos intestinais, diarreia ou obstipação, sangue nas fezes, perda de peso e fadiga constante [2]. Existem vários métodos de deteção e diagnóstico, entre os quais o mais comum é a colonoscopia.

A colonoscopia é um procedimento que usa um tubo estreito, longo e flexível com uma câmara e uma lâmpada no seu extremo, designado por colonoscópio, que permite obter imagens do reto e de largas porções do intestino grosso. Este procedimento pode permitir a visualização de tecido irritado ou inchado, úlceras e pólipos. Estes últimos consistem de tecido extra, que cresce no interior do intestino, e são de particular importância, pois alguns tipos de pólipos podem já ser uma manifestação de cancro ou podem tornar-se cancro [3]. Assim sendo a importância de uma deteção com uma

elevada taxa de sucesso é necessária. No entanto, segundo Douglas K Rex *et al.* [4] é possível que cerca de 24 % dos pólipos possam passar despercebidos ao médico examinador, deste modo, a criação de um método automático de detecção de pólipos com uma maior taxa de sucesso é de grande importância.

Variados métodos automáticos de detecção de pólipos em imagens médicas foram propostos. Miller *et al.* [5] propôs um esquema de classificação híbrido que combina *support vector machines* com um algoritmo genético (GA) para seleção de variáveis. Fu *et al.* [6] aplicou SFS e SFFS (*Sequential Forward Selection* e *Sequential Floating Forward Selection*) para selecionar os vetores de entrada no classificador baseado em SVM. No caso similar de imagens extraídas por cápsula endoscópica, Yuan e Meng [7] propuseram um algoritmo baseado na integração do método BoF (*Bag of Features*) e do mapa de saliências para a detecção de pólipos em imagens médicas. Assim como o presente estudo, grande parte dos trabalhos publicados na área da detecção automática de pólipos assenta na extração de *features* da imagem completa, seguida da aplicação de um método de classificação.

1.2 Objetivos

A presente dissertação tem como principal objetivo a criação de um algoritmo em MATLAB[®] capaz de reconhecer e classificar pólipos em imagens de colonoscopias. O algoritmo utilizará diferentes métodos já desenvolvidos para as diferentes etapas deste processo, o que deverá permitir efetuar um estudo comparativo entre eles. Para alcançar os objetivos propostos, o algoritmo deverá também concretizar com sucesso as diferentes etapas que antecedem a classificação propriamente dita, desde o pré-processamento das imagens digitais, extração das características (*features*) e a construção de um modelo

classificativo robusto.

1.3 Estrutura da Dissertação

Esta dissertação encontra-se organizada em cinco capítulos, estruturados e enumerados da seguinte forma:

- Capítulo 1, **Introdução**: identificação do problema base a esta dissertação, apresentação dos objetivos e da estrutura do documento;
- Capítulo 2, **Conceitos Fundamentais**: apresentação dos conceitos técnicos e teóricos fundamentais subjacentes ao problema em questão. Nesta dissertação irão ser abordados os seguintes conceitos:

- Aquisição e Tratamento de Imagens;
- Extração de Features;
- Criação de Modelos Classificativos;
- Classificação.

Estes conceitos gerais servem como introdução teórica ao trabalho realizado.

- Capítulo 3 **Trabalho Realizado**: apresentação dos passos práticos e principais métodos utilizados no desenvolvimento do sistema proposto, nomeadamente nos seguintes pontos:
 - Pré-Processamento;
 - Extração de Características;
 - Seleção Estatística de *features*;
 - Classificação.

- Capítulo 4 **Resultados**: apresentação e discussão dos resultados obtidos.
- Capítulo 5 **Conclusões**: avaliação do sistema criado e conclusões.

Capítulo 2

Conceitos Fundamentais

Neste capítulo pretende-se tratar alguns dos conceitos teóricos fundamentais para a realização do trabalho. Os temas serão tratados superficialmente, mas referenciando outros trabalhos e estudos onde se poderá efetuar uma leitura mais extensiva dos assuntos tratados.

2.1 Aquisição e Tratamento de Imagens

2.1.1 Colonoscopia

A colonoscopia tradicional consiste muitas vezes na gravação de imagem colorida em vídeo através de uma câmara digital. No presente trabalho foram extraídas *frames*, completas ou parciais, de vários vídeos médicos.

2.1.2 Pré-Processamento

O pré-processamento das imagens digitais é muitas vezes uma etapa importante que antecede o estudo das mesmas imagens, numa tentativa de melhorar os resultados obtidos. Rafael C. Gonzalez e Richard E. Woods abordam no

seu livro *Digital Image Processing* [8] uma série extensiva de técnicas usadas no melhoramento de imagens digitais.

No presente trabalho decidiu-se, após vários estudos, limitar o pré processamento das imagens à simples seleção da região de interesse (ROI). Devido à natureza do sistema utilizado para a obtenção de imagens numa colonoscopia, a imagem resultante tem um formato octogonal não regular o que produz, quando representada num formato $x \times y$ pixels, a ocorrência de cantos sem informação relevante. A presença desses cantos introduz elementos (fronteiras) que não devem ser considerados na criação do modelo classificativo. Deste modo foi criado um sistema baseado em operações de morfologia matemática que permitiu rejeitar os artefactos referidos anteriormente e assim transformar as imagens de modo a que estas sejam apenas compostas pela parte relevante da imagem, ou seja, a ROI da imagem original.

Cor

Como referido anteriormente, as imagens provenientes da colonoscopia são imagens a cores com três canais: vermelho, verde e azul (RGB). Assim como grande parte da literatura sobre a extração de *features*, também este estudo trabalha essencialmente com imagens em escala de cinzentos. No entanto, antes da conversão do formato tridimensional (RGB) para um formato bidimensional (*gray level*) é de interesse extrair e guardar alguma informação que é dada pela presença e distribuição das cores na imagem. Assim pode-se dizer que a extração de *features* elaborada neste trabalho é efetuada em dois momentos distintos quando analisada em função do formato da imagem: **imagens a cores** - extração de *features* baseadas na cor; **imagens em escala de cinzento** - extração de *features* locais e de textura.

2.2 Extração de Features

A extração de *features* é o segundo passo no sistema de classificação proposto, sucedendo à seleção da ROI, e é também um dos mais importantes. A extração e seleção de *features* é fundamental para a construção de um modelo classificativo robusto e com isso em mente subdividiu-se este processo em três partes distintas como se trata de seguida.

2.2.1 Features Baseadas na Cor

Na identificação de um tecido anormal na parede intestinal, a cor pode por vezes fornecer informação relevante. Assim, no presente trabalho foram obtidas 3 características a partir da informação transmitida pela cor da imagem. A tabela 2.1 mostra as características extraídas.

Característica	Formulação
Média da Componente Normalizada de Vermelho	$r = MED \left(\frac{R}{R+G+B} \right)$
Média da Componente Normalizada de Verde	$g = MED \left(\frac{G}{R+G+B} \right)$
Intensidade Média	$i = MED \left(\frac{R+G+B}{3} \right)$

Tabela 2.1: As três características extraídas com base na cor das imagens. R, G e B são matrizes do tamanho da imagem em pixels que indicam qual o valor de cada componente (vermelho, verde e azul) para esse pixel.

2.2.2 Features Locais

As características de natureza local são *features* provenientes de pontos de interesse encontrados nas imagens em escala de cinzentos. Estes pontos de

interesse são normalmente encontrados por vários detetores: detetores de arestas, detetores de cantos ou detetores de manchas (*blobs*). Por outro lado existem vários algoritmos, que de uma forma completa e estruturada, detetam e identificam estes pontos de interesse. Estes algoritmos podem por sua vez utilizar, numa ou várias etapas, alguma forma dos detetores enumerados anteriormente. Neste trabalho estudaram-se e aplicaram-se dois conhecidos algoritmos de extração de *features* locais: SIFT e SURF.

O SIFT (*Scale-Invariant Feature Transform*), implementado por David Lowe em 1999 [9] é um algoritmo amplamente conhecido e a partir do qual vários outros foram derivados, em particular a implementação de A. Vedaldi and B. Fulkerson [10] que é utilizada neste trabalho. Sumariamente, o SIFT assenta na extração de pontos localizados em regiões de alto contraste na imagem, pontos estes que caracterizam a imagem (ou objeto presente na imagem). Como na prática o SIFT extrai um número elevado de pontos torna-se bastante robusto a variações de escala, translações e oclusões parciais.

O SURF (*Speeded Up Robust Features*), apresentado por Herbert Bay *et al.* em 2006 [11] é um algoritmo inspirado no SIFT mas que implementa algumas melhorias, nomeadamente ao nível da rapidez de computação. O SURF assenta principalmente na utilização do determinante da matriz Hessiana como detetor de *blobs*.

2.2.3 Features de Textura

A textura é uma característica importante na identificação de objetos e outras regiões de interesse numa imagem. No presente trabalho foi usada uma rotina em Matlab[®] criada por Avinash Uppuluri que calcula um total de 22 *features* sugeridas por R. M. Haralick *et al.*, L. Soh e C. Tsatsoulis e ainda D A. Clausi,

a partir da matriz de coocorrências em escala de cinzentos.[12][13][14][15]

Assumindo que $p(i, j)$ é a entrada (i, j) da matriz de coocorrências normalizada, N_g é o número de níveis de cinzento e ainda onde μ_x , μ_y , σ_x e σ_y são as médias e desvios padrão por linha e por coluna da matriz de coocorrências, as *features* calculadas foram as seguintes:

- Auto-correlação:

$$f_1 = \sum_i \sum_j (i, j)p(i, j). \quad (2.1)$$

- Contraste:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=i}^{N_g} \sum_{j=1}^{N_g} p(i, j) \mid |i - j| = n \right\}. \quad (2.2)$$

- Correlação Matlab[®]

- Correlação:

$$f_4 = \frac{\sum_i \sum_j (i, j)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}. \quad (2.3)$$

- *Cluster Prominence*:

$$f_5 = \sum_i \sum_j (i + j - \mu_x - \mu_y)^4 p(i, j). \quad (2.4)$$

- *Cluster Shade*:

$$f_6 = \sum_i \sum_j (i + j - \mu_x - \mu_y)^3 p(i, j). \quad (2.5)$$

- Dissimilaridade:

$$f_7 = \sum_i \sum_j |i - j| \cdot (i, j). \quad (2.6)$$

- Energia:

$$f_8 = \sum_i \sum_j p(i, j)^2. \quad (2.7)$$

- Entropia:

$$f_9 = - \sum_i \sum_j p(i, j) \log p(i, j). \quad (2.8)$$

- Homogeneidade Matlab[®]

- Homogeneidade:

$$f_{11} = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j). \quad (2.9)$$

- Probabilidade Máxima:

$$f_{12} = \max_{i,j} p(i, j) \quad (2.10)$$

- Soma de Quadrados: Variância:

$$f_{13} = \sum_i \sum_j (i - \mu)^2 p(i, j). \quad (2.11)$$

- Média da Soma:

$$f_{14} = \sum_{i=2}^{2N_g} i p_{x+y}(i). \quad (2.12)$$

- Variância da Soma:

$$f_{15} = \sum_{i=2}^{2N_g} (i - f_{14})^2 p_{x+y}(i). \quad (2.13)$$

- Entropia da Soma:

$$f_{16} = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log p_{x+y}(i). \quad (2.14)$$

- Variância da Diferença:

$$f_{17} = \text{variância de } p_{x-y}. \quad (2.15)$$

- Entropia da Diferença:

$$f_{18} = - \sum_{i=0}^{N_g-1} p_{x-y} \log p_{x-y}(i). \quad (2.16)$$

- Informação de Medidas de Correlação:

$$f_{19} = \frac{HXY - HXY1}{\max HX, HY}. \quad (2.17)$$

$$f_{20} = ((1 - \exp[-2.0(HXY2 - HXY)])^{\frac{1}{2}}). \quad (2.18)$$

Com $HXY = - \sum_i \sum_j p(i, j) \log p(i, j)$, onde HX e HY são as entropias de p_x e p_y , respetivamente e ainda:

$$\begin{aligned} HXY1 &= - \sum_i \sum_j p(i, j) \log p_x(i, j) p_y(i, j) \\ HXY2 &= - \sum_i \sum_j p_x(i) p_y(j) \log p_x(i, j) p_y(i, j) \end{aligned}$$

- Diferença Inversa:

$$f_{21} = \sum_i \sum_j \frac{p(i, j)}{1 + |i - j|} \quad (2.19)$$

- Momento da Diferença Inversa:

$$f_{22} = \sum_i \sum_j \frac{p(i, j)}{1 + (i - j)^2} \quad (2.20)$$

2.3 Seleção estatística de *features*

O aumento da capacidade de obtenção e armazenamento de dados observada nos últimos anos tem levado a uma sobrecarga da quantidade de informação

em muitas áreas. A grande quantidade de variáveis nas observações realizadas pelos mais variados sistemas de aquisição, mas mais particularmente nos sistemas de aprendizagem (*machine learning*), causa novos desafios aos sistemas computacionais de processamento. No entanto, muitas vezes verifica-se que nem todas as variáveis medidas/calculadas são "importantes" para a compreensão dos fenômenos de interesse e embora existam modelos capazes de construir modelos preditivos de grande precisão a partir de dados com elevada dimensão, continua a ser interessante na maioria das aplicações a redução da dimensão da informação original antes de serem criados modelos preditivos [16].

No presente trabalho, devido à grande quantidade de variáveis que foram medidas e calculadas a partir das imagens, também se achou necessário proceder a uma redução do espaço dimensional destes dados e para isso utilizou-se a técnica de PCA (*Principal Component Analysis*), uma técnica conhecida e bem estabelecida, que produz os melhores resultados, no sentido dos mínimos quadrados, na redução linear da dimensão dos dados. Na sua essência, o PCA procura a redução da dimensão dos dados encontrando as poucas combinações lineares ortogonais das variáveis originais que possuam a maior variância. A aplicação do PCA a um conjunto largo de dados pode ser resumida nos seguintes passos [17]:

1. Obtenção dos dados: os dados encontram-se na forma de uma matriz, com uma observação por linha e onde cada coluna é uma variável;
2. Subtração da média: a média deve ser subtraída para centrar os dados;
3. Cálculo da matriz de covariância: a matriz de covariância, que é a generalização da variância para dados multidimensionais, será uma matriz quadrada com $(N_{vars})^2$ elementos, onde N_{vars} é o número de variáveis;

4. Cálculo dos valores e vetores próprios: o número de valores e vetores próprios é igual ao número de variáveis. Os vetores próprios indicam as direções segundo as quais os dados podem ser expressos, por outro lado, os valores próprios indicam a "certeza" relativa com que cada vetor próprio caracteriza os dados na sua generalidade. Por exemplo, vejamos o caso bidimensional onde se calcularam os valores e vetores próprios: se o segundo valor próprio for muito mais elevado que o outro indica que os dados são muito melhor explicados pelo segundo vetor próprio, como se os dados estivessem dispostos numa elipse alongada. Se os valores próprios forem semelhantes em magnitude pode-se considerar que nenhum dos vetores explica particularmente bem os dados, como é o caso em que estes dados estejam dispostos na forma de um círculo;
5. Seleção de componentes: como visto atrás, o maior valor próprio é o que melhor explica os dados, como tal este é a *componente principal*. Para seleccionar os dados deverá então ordenar-se os valores próprios por ordem decrescente, e escolher os p primeiros dos N_{vars} totais: isto tem como consequência direta a redução da dimensão dos dados, passando de N_{vars} dimensões para p dimensões. O número p de componentes que se seleccionam é relativo, e pode ser definido à partida para um número fixo de componentes, ou ajustado dinamicamente de modo a representar uma percentagem pretendida dos dados originais;
6. Obtenção dos dados finais: os dados finais são o resultado da multiplicação à esquerda da matriz formada pelos vetores próprios escolhidos transposta, pela matriz que contém os dados originais com a média subtraída, também transposta, ou seja:

$$DadosFinais = VetoresProprios^T \times DadosOriginaisCentrados^T$$

Os dados finais, na forma de uma matriz $N_{obs} \times p$ onde N_{obs} indica o número de observações e p é o número de variáveis escolhidas no processo anterior, são então a representação reduzida dos dados originais, obtida pela técnica PCA.

2.4 Classificação

A classificação é o processo final no sistema criado. Esta etapa consiste na análise de novas *features* de acordo com o modelo classificativo criado anteriormente. De acordo com a técnica usada para a criação do modelo é escolhida a técnica de classificação. No caso de *Support Vector Machines* os novos exemplos/*features* são então mapeados para o mesmo espaço que o modelo representa e com este mapeamento é determinada qual a categoria a que determinado exemplo pertence verificando qual a localização final do exemplo mapeado. No que toca a *Classification Trees* os novos exemplos são avaliados segundo a árvore criada no modelo classificativo, ramo a ramo até determinar a "etiqueta" que melhor se adequa.

Capítulo 3

Trabalho Realizado

3.1 Imagens e Estrutura do Sistema

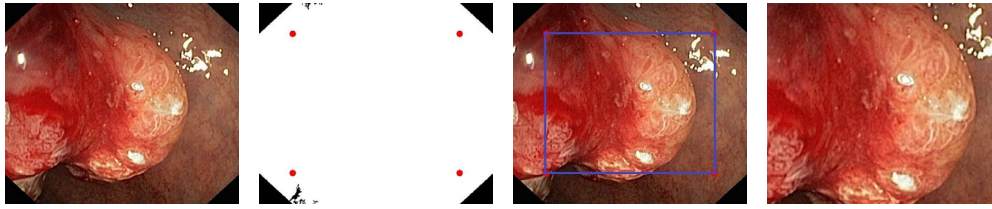
Normalmente, a análise de imagens médicas é uma tarefa realizada por um médico, que manualmente examina todas as imagens num processo visual repetitivo e sujeito a interpretações erradas, mesmo para os médicos mais experientes. O que esta tese propõe é um sistema de classificação automático que possa auxiliar o médico na inspeção das imagens e ajude a aumentar a taxa de sucesso deste processo. Neste trabalho foram utilizadas 201 imagens, compostas de 40 casos onde ocorrem pólipos benignos, 75 imagens com pólipos malignos e 86 imagens sem nenhuma ocorrência.

O sistema proposto é constituído pelas etapas de extração de features, aprendizagem e classificação. Em primeiro lugar uma coleção de features são retiradas das imagens. Estas são compostas por 3 variáveis associadas à cor da imagem, um número variável de features locais obtidos pelo método SURF (*Speeded-Up Robust Features*) ou pelo método SIFT *Scale-Invariant Feature Transform* implementado por [10] na sua biblioteca de código aberto VLFeat [10]. As restantes features são obtidas a partir da textura da imagem tirando

partido da matriz de coocorrências. Este sistema propõe que de seguida esse conjunto de features seja utilizado por um de dois métodos de aprendizagem: árvores de classificação (*Classification Trees*) ou vetores de suporte (*Support Vector Machines*) para a criação de um modelo de classificação. O modelo de classificação criado é então utilizado para classificar novas imagens como sendo de um de três tipos distintos: contêm pólipo maligno, contêm pólipo benigno e não contêm pólipos.

3.2 Pré-Processamento

Devido à natureza do sistema de aquisição de imagens estas possuem bordos, ou pelo menos cantos, que não contêm informação relevante para a análise a efetuar como se pode verificar na imagem 3.1(a). Como estes artefactos introduzem uma fronteira fictícia que afeta a extração de características, torna-se necessário definir a região de interesse (*ROI - Region of Interest*) como sendo a porção da imagem que fica no interior de um retângulo sobreposto à imagem que não inclua os bordos a negro. Para definir esta ROI é criada uma matriz binária com um número de elementos igual ao número de pixels da imagem. Nesta matriz, encontram-se a 1 (um) todos os elementos que correspondem à parte colorida da imagem e a 0 (zero) os elementos que correspondem aos bordos a negro como mostra a figura 3.1(b). De seguida, encontram-se os índices do primeiro e último elemento de cada lateral da imagem que irão definir os quatro cantos da ROI. Finalmente a imagem inicial é cortada de acordo com a região definida 3.1(d).



(a) Exemplo de uma imagem obtida pelo sistema de aquisição ilustrados. (b) Matriz binária com vértices da ROI ilustrados. (c) Imagem original com ROI ilustrada. (d) Imagem final após corte da zona exterior à ROI.

Figura 3.1: Sequência de passos da definição da região de interesse.

3.3 Extração de Características

As características, ou *features*, obtidas podem ser separadas em três grupos distintos: as *features* baseadas na cor da imagem; as *features* locais, obtidas a partir de técnicas bem estabelecidas como o SURF e o SIFT; e ainda as *features* de textura, calculadas a partir da matriz de coocorrências da imagem. Todas estas características são agrupadas numa matriz onde cada linha corresponde a uma imagem e cada coluna é uma variável (p.e. componente normalizada de vermelho). As secções seguintes abordam como cada grupo de *features* foi obtida.

3.3.1 Características Baseadas na Cor

Como já foi referido, as imagens utilizadas são imagens digitais com três componentes de cor, neste caso RGB. Neste sistema decidiram-se extrair três características a partir dessas componentes de cor: as componentes normalizadas r e g referentes às cores vermelha e verde e ainda a intensidade média da imagem. Assim, para cada imagem, os três valores referentes a estas

features constituem os primeiros elementos da sua linha correspondente na matriz de features final.

3.3.2 Features Locais

SURF

No presente trabalho foram adotadas duas técnicas, SURF e SIFT, para a obtenção de *features* de natureza local. As *features* SURF foram obtidas através da função *detectSURFFeatures* da *toolbox Computer Vision System* do MATLAB[®]. Os pontos de interesse devolvidos são então reduzidos aos dez mais fortes utilizando a função *selectStrongest*. Por fim as *features* são extraídas utilizando a função *extractFeatures*. O resultado final deste processo é uma matriz de 10 linhas por 64 colunas, onde cada linha é uma *feature* de 64 variáveis (colunas). Este processo pode ser resumido da seguinte forma:

1. É chamada a função *detectSURFFeatures* com a imagem como único parâmetro de entrada:
 - A imagem que é passada como parâmetro está em escala de cinzentos de 8 bits.
 - É devolvida uma estrutura da classe *SURFPoints* que contém todos os pontos de interesse encontrados.
2. É chamada a função *selectStrongest* que tem como parâmetros a estrutura anterior e ainda o número de pontos que se pretende:
 - Para este trabalho optou-se por apenas preservar os dez pontos mais fortes.
 - Esta rotina devolve uma estrutura semelhante à anterior, mas apenas com dez elementos.

3. Utiliza-se a função *extractFeatures* para extrair as features a partir do conjunto de dez pontos de interesse:
 - Esta função devolve uma matriz 10x64 onde cada linha é uma *feature* e cada coluna é uma variável.
4. Esta matriz é transformada de forma a ser incorporada no restante algoritmo. Na prática esta transformação reduz as 10 *features* a apenas uma, que engloba todas as características de natureza local:
 - O formato final é um vetor linha de 10x64=640 valores.
5. O vetor anterior é concatenado ao vetor de 3 elementos (*features* de cor) já existente, formando um vetor de 643 elementos. Na secção 3.3.3 vamos ver como são obtidos os restantes elementos que formam a *feature* final para a imagem.

Como é natural, o número de pontos de interesse varia para cada imagem, como tal o algoritmo acima referido pode na prática devolver menos de 10 pontos de interesse. Neste caso não é possível criar um vetor de 640 elementos diretamente a partir das *features* encontradas pelo SURF, pelo que se torna necessário preencher o resto do vetor com zeros de modo a respeitar as dimensões assumidas nas restantes partes do sistema criado.

SIFT

Anteriormente vimos como são obtidas as *features* locais utilizando o método SURF, no entanto neste trabalho também foi implementado e testado um algoritmo independente mas semelhante que usa a técnica SIFT como método de obtenção das *features* locais das imagens médicas. A. Vedaldi and B. Fulkerson [10] desenvolveram uma série de ferramentas que permitem

a implementação desta técnica com facilidade, das quais foi utilizada a rotina *vl_sift* que recebe como parâmetro uma imagem *graylevel* e devolve um conjunto de pontos de interesse e respectivos descritores. Neste caso estamos interessados na matriz de descritores, ou *features*, devolvida: esta é uma matriz onde cada coluna é uma *feature*, e por sua vez cada *feature* é um vetor de 128 elementos. Como acontece no caso do SURF, o número de pontos de interesse e consequentes *features* varia de imagem para imagem, exceto que neste caso não existe uma rotina nativa que permita restringir à partida o número de pontos de interesse a um conjunto limitado formado pelos mais fortes (como fazia a função *selectStrongest* no caso do SURF). Deste modo todas as *features* são preservadas durante esta fase até à aplicação da PCA (secção 3.4). De qualquer maneira, e como se achou necessário assumir uma postura de redução do tempo de computação, todos vetores e matrizes de dimensão considerável são declarados antes de serem utilizados, assim, existiu a necessidade de determinar qual o tamanho que a matriz teria de possuir de modo a preservar a maioria das *features* devolvidas pelo algoritmo. Assim chegou-se ao compromisso de limitar o vetor total a 15000 elementos o que na prática limita o número de descritores locais a 14975.

Por fim, e como feito para o SURF, este vetor foi concatenado ao vetor já existente com as *features* de cor.

3.3.3 *Features* de Textura

Como referido na secção 2.2.3 foi utilizada uma rotina implementada por Avinash Uppuluri para calcular as *features* de textura. Estas foram obtidas em três passos:

1. Chamada da função Matlab[®] *graycomatrix* com uma imagem em escala de cinzentos como parâmetro. Esta função devolve a matriz de

coocorrências associada a essa imagem.

2. Chamada da função *GLCM_Features*[12] para, por fim, calcular e devolver as *features* associadas à imagem. Esta função tem como parâmetro a matriz de coocorrências calculada no passo anterior e devolve uma estrutura de *features*.
3. Os elementos da estrutura são dispostos num vetor linha que é concatenado ao vetor já existente, que contém as *features* de cor e de natureza local para a imagem em questão.

Neste momento encontra-se criado o vetor de *features* para uma imagem. Este vetor contém 3 variáveis ligadas à cor, um número variável de descritores dos pontos de interesse encontrados e finalmente 22 valores caracterizadores da textura da imagem, e é o vetor que descreve cada imagem neste trabalho. Estes passos de obtenção de *features* são repetidos para todas as imagens já catalogadas e presentes na base de dados, e por fim obtêm-se 201 vetores (número de imagens na base de dados) de *features* que são agrupados numa matriz única. É esta matriz que irá ser utilizada para efetuar a aprendizagem.

3.4 Seleção estatística de *features*

Devido à elevada quantidade de *features* extraídas das imagens, que após construção da matriz final se traduziram em matrizes de 201×665 para o caso em que se utilizou o SURF e 201×15000 para o caso do SIFT, tornou-se necessário reduzir a dimensão da matriz de *features*. Neste trabalho em particular escolheu-se reduzir os dados para 4 dimensões ou variáveis utilizando a técnica PCA como indicado na secção 2.3. Da tabela 3.1 é possível verificar que existe uma "direção" que representa os dados com muito mais importân-

Componente	Valor Próprio
1 ^a	222.347
2 ^a	77.8302
3 ^a	7.08938
4 ^a	3.60965
5 ^a	0.27180
6 ^a	0.13473

Tabela 3.1: Os 6 valores próprios com maior magnitude obtidos aquando da aplicação do PCA a um conjunto de dados com 201 observações e 665 variáveis.

cia do que os restantes, de qualquer modo optou-se por utilizar os quatro primeiros componentes de modo a minimizar a quantidade de informação perdida sem aumentar a complexidade da estrutura em demasia.

Assim, após aplicação do PCA a matriz de *features* atual tem a dimensão 201×4 o que a torna consideravelmente menor que a matriz original, sendo que a informação original permanece quase completamente inalterada (no sentido de preservar quais as direções segundo as quais a variância dos dados é maior).

3.5 Classificação

O processo de classificação está dividido em três etapas distintas: em primeiro lugar, a aprendizagem das *features* obtidas a partir das imagens conhecidas e identificadas, em segundo lugar, a obtenção de novas *features* a partir de novas imagens não-identificadas e que se pretendem catalogar consoante a presença e as características dos pólipos e finalmente a classificação

propriamente dita dessas novas *features* com base na aprendizagem efetuada inicialmente.

3.5.1 Aprendizagem

No presente trabalho foram testadas duas formas de aprendizagem independentes. Uma delas foi uma abordagem baseada em *Support Vector Machines*. Nesta abordagem, fez-se uso da função *svmtrain* da *toolbox* de estatística do Matlab[®]. O método utilizado na separação dos hiperplanos dos dados foi o método SMO (*Sequential Minimal Optimization*) que é um algoritmo utilizado em programação quadrática e que é largamente utilizado no treino SVM[18]. A função *svmtrain* recebe como parâmetros a matriz de *features* final como descrito na secção 3.4 e ainda um vetor de "etiquetas" que descrevem cada imagem da base de imagens conhecidas relativamente à presença e/ou características dos pólipos. A tabela 3.2 indica sucintamente quais são essas etiquetas e o seu significado.

Finalmente, *svmtrain* devolve uma estrutura caracterizadora das *features* em função das "etiquetas" identificadas. Entre outros, um dos membros dessa estrutura são então os "vetores de suporte", que são uma matriz onde cada linha corresponde a um "vetor de suporte" no espaço dos dados. Esta matriz é uma sub-matriz da matriz de *features* original.

Independentemente foi também testada a prestação das árvores de classificação *Classification Trees* na criação do modelo de classificação. Para esse efeito foi utilizada a rotina *ClassificationTree.fit* também da *toolbox* de estatística. Mais uma vez esta rotina recebeu como únicos parâmetros a matriz de *features* reduzida obtida em 3.4 e o vetor de "etiquetas". O resultado é devolvido por esta rotina é uma estrutura caracterizadora dos dados iniciais.

Tanto SVM como as árvores de classificação são de natureza binária, no

Etiqueta	Significado
Benigno	A imagem contém um ou mais pólipos que foi identificado como não sendo imediatamente perigoso.
Normal	A imagem não contém estruturas anormais.
Maligno	A imagem contém um ou mais pólipos que constituem tecido canceroso.

Tabela 3.2: Significado das três etiquetas usadas para identificação das imagens.

entanto, um dos objetivos do presente trabalho era classificar imagens em três tipos distintos, portanto, tanto para um método como para o outro a etapa de aprendizagem foi efetuada em dois níveis distintos. Num primeiro nível o objetivo foi classificar imagens como sendo "Maligno" ou "outro" e num segundo nível, efetuar uma segunda aprendizagem binária a aplicar apenas às imagens etiquetadas como "Benigno" ou "Normal". Deste modo tornando possível a identificação de três tipos distintos utilizando apenas decisores binários.

3.5.2 Extração de novas *features*

Para a classificação de novas imagens é necessária a extração das suas *features*. Este ponto consiste na aplicação de todos os passos falados nas secções 3.2 e 3.3. De seguida a redução da dimensão das *features* é efetuada utilizando os vetores próprios obtidos na secção 3.4 uma vez que se considerou que o número de observações iniciais foram suficientes.

Assim tem-se uma matriz de *features* cujo número de linhas é o número de imagens que se pretendem classificar e as colunas são as quatro componentes principais segundo as quais esta classificação irá ser efetuada. Na prática,

e como irá ser visto na discussão dos resultados, não foram obtidas novas *features* mas sim utilizadas as atuais recorrendo ao método *leave-one-out*.

3.5.3 Classificação

Para cada uma das técnicas de aprendizagem utilizadas existe uma técnica complementar de classificação. Neste trabalho utilizou-se uma abordagem *standard* do ponto de vista do algoritmo, onde a classificação foi efetuada recorrendo à função *svmclassify* como complemento da *svmtrain* utilizada na aprendizagem; e à *predict* no caso em que se utilizaram árvores de classificação na criação do modelo.

3.6 Resumo

Esta secção pretende indicar, de forma descritiva e sucinta, como o trabalho foi realizado através de itens numerados. Antes de iniciar o processo assume-se o seguinte:

- As imagens já conhecidas estão separadas em três pastas distintas consoante a sua "etiqueta": "benigno", "maligno" e "normal" são as três pastas que contêm todas as imagens obtidas *a priori* e classificadas manualmente;
- O compilador Matlab[®] (MCR) disponível gratuitamente nas páginas web da Mathworks está instalado e funcional;
- A biblioteca VL_Feat está disponível na máquina local e o seu diretório é conhecido;

Tendo em consideração o anterior, o processo de classificação de imagens pode ser decomposto nos seguintes passos:

1. Chamada da rotina *CaracteristicasDeImagens*: esta rotina está encarregue de criar as matrizes para armazenamento de *features*, de seguida percorre as todas as imagens extraíndo as *features* de cor, locais e de textura para cada uma. É então aplicado o PCA à matriz de *features* que reduz a matriz final. Por fim, cria o vetor de "etiquetas" que possui um elemento em texto para identificar cada imagem. As *features*, os vetores próprios calculados durante o PCA e o vetor de etiquetas são devolvidos numa estrutura;
2. Chamada da rotina que efetua a aprendizagem:
 - *TreinoSVM* para uma aprendizagem baseada em vetores de suporte;
 - *ClassificationTree.fit* para uma aprendizagem baseada em árvores de classificação;

Em ambos os casos são utilizados como parâmetros a matriz de *features* e o vetor de "etiquetas" e é devolvida uma estrutura que representa o modelo classificativo implementado.

3. Chamada da rotina *ExtrairFeaturesUnicaImagem*: esta rotina é chamada uma vez por cada imagem que se pretende classificar e a matriz de *features* é "montada" uma linha por imagem. No final esta matriz é centrada (subtração da média) e é multiplicada pelos vetores próprios obtidos em 1. .
4. Classificação utilizando:
 - *ClassificarImagens* para efetuar uma classificação utilizando a estrutura criada por *TreinoSVM*;

- *predict* para efetuar uma classificação utilizando a estrutura criada por *ClassificationTree.fit*;

Em ambos os casos, o resultado desta classificação é devolvido em forma de vetor, com um elemento por cada imagem, em que cada elemento é a etiqueta encontrada no processo de classificação para identificar a respectiva imagem.

Capítulo 4

Resultados

O *software* da presente dissertação foi desenvolvido e testado em Matlab[®] R2013a num PC Intel[®] Core[™]i5-2410M CPU @ 2.30GHz com 6,00GB de RAM instalada. A análise da performance é feita numa primeira fase através da consideração de tabelas de taxas de sucesso para as três categorias e de taxas de falsos positivos e falsos negativos para a situação binária equivalente. Este problema binário é derivado do anterior considerando os resultados "benigno" e "normal" como diagnósticos negativos e "maligno" como um diagnóstico positivo.

As tabelas foram criadas de modo a permitir uma comparação rápida entre os os vários aspetos abrangidos: a comparação entre dois métodos de obtenção de *features* locais (SIFT e SURF) e entre dois métodos de classificação (SVM e Árvores Binárias).

De modo a obter valores mais representativos optou-se pela não utilização de novas imagens, mas sim pela utilização do método *leave-on-out* em todas as 201 imagens disponíveis na altura, maximizando assim o número de classificações efetuadas sem diminuir a quantidade de imagens para aprendizagem.

		Método de Classificação					
		SVM			Árvores Binárias		
		VB	VN	VM	VB	VN	VM
Método de obtenção de features locais	SURF	55%	50%	82.66%	80%	93%	96%
	SIFT	52.5%	50%	85.33%	80%	93%	93.33%

Tabela 4.1: Tabela dos resultados de classificação em três categorias. VB, VN e VM são as taxas de classificação correta dos casos benignos, normais e malignos, respetivamente.

As tabelas 4.2 até 4.5 representam as matrizes de confusão para todas as situações testadas. A partir destas tabelas de contagem básicas é possível extrair um número de métricas importantes para a classificação do modelo criado. A tabela 4.6 representa uma dessas métricas: as taxas de falsos positivos e falsos negativos.

		Condição	
		Maligno	Não Maligno
Resultado do teste	Maligno	72	8
	Não Maligno	3	118

Tabela 4.2: Matriz de confusão para SURF/Árvores Binárias.

		Condição	
		Maligno	Não Maligno
Resultado do teste	Maligno	62	25
	Não Maligno	13	101

Tabela 4.3: Matriz de confusão para SURF/SVM.

		Condição	
		Maligno	Não Maligno
Resultado do teste	Maligno	70	7
	Não Maligno	5	119

Tabela 4.4: Matriz de confusão para SIFT/Árvores Binárias.

		Condição	
		Maligno	Não Maligno
Resultado do teste	Maligno	64	26
	Não Maligno	11	100

Tabela 4.5: Matriz de confusão para SIFT/SVM.

Outras métricas que são importantes calcular são a Precisão e o Reconhecimento (*Recall*). A tabela 4.7 mostra os valores obtidos para estas duas métricas com base nas tabelas de confusão. No presente contexto, a precisão indica a probabilidade de um resultado Maligno ser realmente Maligno, enquanto que o *Recall* indica a probabilidade de um caso Maligno ser identificado como tal. Tendo em conta esta definição é possível observar que o *Recall* pode ser aumentado indiscriminadamente até ao máximo de 100%, situação em que todos os casos Malignos são identificados como tal.

No entanto, o afinamento do sistema para ser demasiado sensível, pode provocar um decréscimo da precisão, aumentando o número de casos Não

		Método de Classificação			
		SVM		Árvores Binárias	
		FP	FN	FP	FN
Método de obtenção de features locais	SURF	19.84%	17.33%	6.34%	4%
	SIFT	14.6%	20.63%	5.55%	6.67%

Tabela 4.6: Taxas de falhas num classificação binária. FP e FN representam a percentagem de falsos positivos e de falsos negativos, respetivamente.

Malignos que são identificados erradamente como Malignos. Embora esta situação seja desejável para excluir definitivamente a hipótese de um pólipso Maligno aquando de uma identificação Negativa, a identificação positiva pode ser inconclusiva caso a precisão não seja suficientemente elevada.

		Método de Classificação			
		SVM		Árvores Binárias	
		<i>Precision</i>	<i>Recall</i>	<i>Precision</i>	<i>Recall</i>
Método de obtenção de features locais	SURF	71%	83%	90%	96%
	SIFT	90%	85%	91%	93%

Tabela 4.7: Precisão e *Recall* obtidos utilizando aquelas que provaram ser as melhores condições do sistema. O modelo baseado em árvores binárias é alguns pontos percentuais superior ao baseado em SVM em todas as métricas testadas e para ambos os algoritmos de obtenção de *features* locais.

Para a que se achou ser a melhor combinação de métodos, SURF e Árvores Binárias, foi criada também a curva *Precision vs Recall* mostrada na Imagem 4.1.

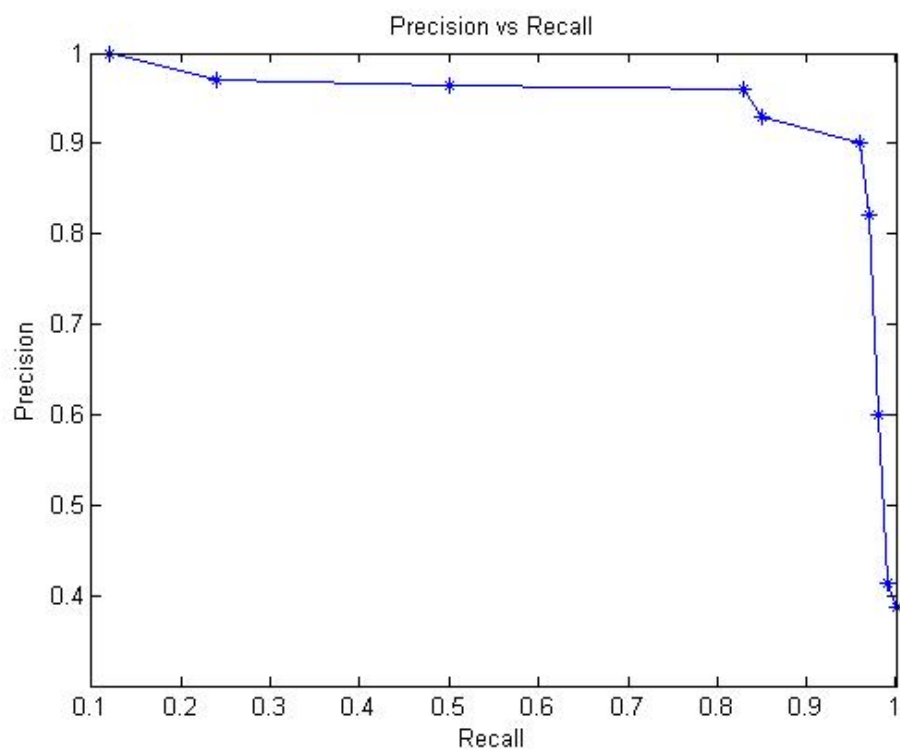


Figura 4.1: Curva *Precision vs Recall* para o sistema com os métodos SURF e Árvores Binárias.

Capítulo 5

Conclusões

Este trabalho propôs e comparou vários métodos de criação de um sistema de detecção de pólipos em imagens de colonoscopias. Essas imagens foram melhoradas através da seleção da região de interesse de modo a eliminar fronteiras artificiais não relevantes. De seguida, procedeu-se à extração de *features* de natureza local, com base na cor e com base na textura dessas mesmas imagens. A maior parte dessas *features* foram de natureza local, fornecidas pelo método SIFT ou SURF. Ao conjunto das *features* extraídas foi aplicada uma seleção estatística, relativamente às suas componentes principais, de modo a reduzir a sua dimensão e facilitar a sua utilização computacional. Por fim, esta matriz reduzida foi utilizada para classificar imagens de acordo com dois métodos distintos: SVM e árvores de classificação. De modo a testar esta classificação foi utilizado o método *leave-one-out* de modo a maximizar o número de classificações efetuadas.

Devido à preponderância que as *features* de natureza local possuem, esses foram um dos parâmetros para a criação de tabelas de performance. Os resultados obtidos (ver tabelas do Capítulo 4) refletem as variações entre, quer os métodos de obtenção de características locais como referido anterior-

mente, quer os métodos de classificação empregues. É necessário admitir que o universo de cerca de duzentas imagens de aprendizagem pode ser reduzido o que pode dar aso ao aparecimento de artefactos estatísticos. No entanto, algumas conclusões podem ser retiradas com confiança:

- o algoritmo classificativo baseado em árvores binárias é consistentemente mais preciso e com menores taxas de classificações erradas, quer na classificação em três categorias, quer na classificação binária;
- o algoritmo SVM, com o *setup* utilizado neste trabalho, tem algumas dificuldades em distinguir entre casos Benignos e Normais, chegando até a ser apenas 20% melhor que uma classificação aleatória (33% de hipóteses de um sistema aleatório acertar um diagnóstico em três);
- devido aos dois pontos anteriores este trabalho sugere que o método baseado em árvores binários como o método mais adequado, ainda para mais tendo em conta que a aplicação prática das árvores binárias foi de uma complexidade semelhante à implementação do método SVM;
- os métodos de obtenção de *features* locais produzem resultados bastante semelhantes entre si, sendo que o método SIFT foi de uma implementação prática muito mais complexa e exigiu um maior esforço computacional. Assim, este trabalho sugere que o método SURF será o mais adequado para desenvolvimentos futuros.

Bibliografia

- [1] Iarc database. 2012 lyon, france globocan. http://globocan.iarc.fr/Pages/fact_sheets_population.aspx. Consultado 2014 Abr 15.
- [2] Pop - portal de oncologia portugues. 2014 portugal. <http://www.pop.eu.com/portal/publico-geral/tipos-de-cancro/cancro-colo-rectal/sinais-e-sintomas-colo-rectal.html>. Consultado 2014 Abr 15.
- [3] The national institute of diabetes and digestive and kidney diseases, bethesda, usa. <http://digestive.niddk.nih.gov/ddiseases/pubs/colonoscopy/>. Consultado 2014 Abr 16.
- [4] DOUGLAS K Rex, CHRISTOPHER S Cutler, GREGORY T Lemmel, EMAD Y Rahmani, DAVID W Clark, DEBRA J Helper, GLEN A Lehman, and DAVID G Mark. Colonoscopic miss rates of adenomas determined by back-to-back colonoscopies. *Gastroenterology*, 112(1):24–28, 1997.
- [5] Meghan T Miller, Anna K Jerebko, James D Malley, and Ronald M Summers. Feature selection for computer-aided polyp detection using genetic algorithms. In *Medical Imaging 2003*, pages 102–110. International Society for Optics and Photonics, 2003.

- [6] Jachih JC Fu, Ya-Wen Yu, Hong-Mau Lin, Jyh-Wen Chai, and Clayton Chi-Chang Chen. Feature extraction and pattern classification of colorectal polyps in colonoscopic imaging. *Computerized Medical Imaging and Graphics*, 38(4):267–275, 2014.
- [7] Yixuan Yuan and Max Q-H Meng. Polyp classification based on bag of features and saliency in wireless capsule endoscopy.
- [8] Rafael C Gonzales and Richard E Woods. *Digital Image Processing, 2-nd Edition*. Prentice Hall, 2002.
- [9] David G Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision*, volume 2, pages 1150–1157. Ieee, 1999.
- [10] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [11] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.
- [12] Avinash Uppuluri. Gcm texture features. <http://www.mathworks.com/matlabcentral/fileexchange/22354-gcm-features4-m-vectorized-version-of-gcm-features1-m-with-code-changes>, 2010. Consultado 2014 Abr 30.
- [13] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

- [14] L-K Soh and Costas Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *Geoscience and Remote Sensing, IEEE Transactions on*, 37(2):780–795, 1999.
- [15] David A Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of remote sensing*, 28(1):45–62, 2002.
- [16] Imola K Fodor. A survey of dimension reduction techniques, 2002.
- [17] Lindsay I Smith. A tutorial on principal components analysis. *Cornell University, USA*, 51:52, 2002.
- [18] John Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

