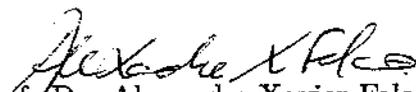


Segmentação de Imagens de Parasitos Intestinais do Homem

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Celso Tetsuo Nagase Suzuki e aprovada
pela Banca Examinadora.

Este exemplar corresponde à redação final da Tese/Dissertação devidamente corrigida e defendida por: <u>Celso Tetsuo Nagase</u> <u>Suzuki</u>
e aprovada pela Banca Examinadora. Campinas, <u>15</u> de <u>maio</u> de <u>2007</u>
 COORDENADOR DE PQS-GRADUAÇÃO CPG-4C

Campinas, 27 de fevereiro de 2007.


Prof. Dr. Alexandre Xavier Falcão
Instituto de Computação - Unicamp
(Orientador)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**
Bibliotecária: Miriam Cristina Alves – CRB8a / 5094

Suzuki, Celso Tetsuo Nagase
Su98s Segmentação de imagens de parasitos intestinais do homem / Celso Tetsuo Nagase Suzuki -- Campinas, [S.P. :s.n.], 2007.
Orientador : Alexandre Xavier Falcão
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.
I. Processamento de imagens. 2. Imagens - Interpretação. 3. Parasito - Análise. I. Falcão, Alexandre Xavier. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.
<p style="text-align: right;">(mca/imecc)</p>

Título em inglês: Image segmentation of human intestinal parasites.

Palavras-chave em inglês (Keywords): 1. Image processing. 2. Image analysis. 3. Parasites – Analysis.

Área de concentração: Processamento e Análise de imagens.

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Alexandre Xavier Falcão (IC-Unicamp)
Prof. Dr. Nelson Delfino d'Ávila Mascarenhas (DC-UFSCar)
Prof. Dr. Ricardo da Silva Torres (IC-Unicamp)

Data da defesa: 27/02/2007

Programa de Pós-Graduação: Mestrado em Ciência da Computação.

Segmentação de Imagens de Parasitos Intestinais do Homem

Celso Tetsuo Nagase Suzuki¹

Fevereiro de 2007

Banca Examinadora:

- Prof. Dr. Alexandre Xavier Falcão
Instituto de Computação - Unicamp (Orientador)
- Prof. Dr. Nelson Delfino d'Ávila Mascarenhas
Departamento de Computação - UFSCar
- Prof. Dr. Ricardo da Silva Torres
Instituto de Computação - Unicamp
- Prof. Dr. Neucimar Jerônimo Leite (Suplente)
Instituto de Computação - Unicamp

¹Suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (processo 04/11218-8).

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 27 de fevereiro de 2007, pela Banca examinadora composta pelos Professores Doutores:

Nelson Delfino D'Ávida Mascarenhas

Prof. Dr. Nelson Delfino D'Ávida Mascarenhas
Departamento de Computação - UFSCar

Ricardo da Silva Torres

Prof. Dr. Ricardo da Silva Torres
IC - UNICAMP.

Alexandre Xavier Falcão

Prof. Dr. Alexandre Xavier Falcão
IC - UNICAMP.

5007 23414

Resumo

Segmentação é o processo de particionar uma imagem em regiões relevantes para uma dada aplicação. A segmentação constitui etapa fundamental de sistemas de reconhecimento de padrões, sendo responsável pela extração de objetos cujas características são usadas para o reconhecimento.

Neste trabalho nós desenvolvemos um método de segmentação, com alta sensibilidade e especificidade, para separar automaticamente os parasitos intestinais do Homem (protozoários e helmintos) do fundo da imagem e de impurezas presentes, como detritos, resíduos alimentares, células animais e vegetais. A base de imagens utilizada consiste em parasitos intestinais de maior prevalência no Brasil. O método proposto alcançou uma sensibilidade diagnóstica de 97,36% na detecção de parasitos, com o uso de um novo classificador de padrões baseado em floresta de caminhos ótimos.

Este trabalho é uma contribuição para o desenvolvimento de um sistema de diagnóstico automático de parasitos intestinais do Homem, para reduzir significativamente a taxa de erro no diagnóstico por inspeção humana.

Abstract

Segmentation is a process to partition an image into relevant regions for a given application. Segmentation is a fundamental stage of pattern recognition systems, being responsible for the extraction of objects whose features are used for recognition.

In this work we developed a segmentation method, with high sensitivity and specificity, to automatically separate human intestinal parasites (protozoa and helminth) from the background and impurities, like debris, food residue, animal and vegetal cells. The image collection used consists of intestinal parasites with high prevalence in Brazil. The proposed method reached a diagnostic sensitivity of 97,36% in detecting parasites, using a new pattern classifier based on optimum-path forest.

This work is a contribution to the development of an automatic diagnostic system for human intestinal parasites, to significantly reduce the error rate in diagnostic by human inspection.

Agradecimentos

Agradeço ao Prof. Dr. Alexandre Xavier Falcão, pela dedicação na orientação, aos companheiros Jancarlo Ferreira Gomes e João Paulo Papa pela constante dedicação ao projeto, à minha namorada e jornalista Ligia Gielamo Oliveira pela compreensão e revisão do texto, à minha família e a todas as outras pessoas que me apoiaram e acompanharam este trabalho.

Agradeço também à FAPESP pelo apoio financeiro e à Agência de Inovação Inova Unicamp pela gestão da patente criada.

Sumário

Resumo	v
Abstract	vi
Agradecimentos	vii
1 Introdução	1
1.1 Motivação do projeto principal	2
1.2 Objetivos	2
1.3 Contribuições	3
1.4 Organização do trabalho	3
2 Fundamentos de Processamento de Imagens	4
2.1 Imagem Digital	4
2.1.1 Imagem em tons de cinza	4
2.1.2 Imagem binária	4
2.1.3 Imagem multibanda	5
2.1.4 Amostragem e quantização	5
2.2 Topologia Digital	5
2.2.1 Relação Binária	5
2.2.2 Métrica	5
2.2.3 Relação de Adjacência e Grafos	6
2.2.4 Relação de Conexidade	6
2.2.5 Componente Conexo	6
2.3 Transformada Imagem-Floresta	7
2.3.1 Definições	7
2.3.2 IFT	8
2.3.3 Aplicações	10
2.4 Realce de bordas	11

3	Fundamentos de Reconhecimento de Padrões	13
3.1	Classificador Rede Neural	14
3.2	Classificador Máquina de Vetores de Suporte	18
3.3	Classificador Floresta de Caminhos Ótimos	21
3.3.1	Algoritmo de aprendizado	23
3.4	Coleção de classificadores	26
3.4.1	<i>Bagging</i>	26
3.4.2	<i>Adaboost</i>	28
4	Base de Imagens	30
4.1	Aquisição das imagens	30
4.2	Parasitas	31
4.3	Bases de dados para projeto e teste	35
5	Segmentação de imagens	36
5.1	Apresentação do problema	36
5.2	Descrição do Método	36
5.2.1	Delineamento de componentes	38
5.2.2	Reconhecimento	46
6	Experimentos e resultados	49
6.1	Resultados do delineamento	49
6.2	Treinamento dos classificadores	50
6.3	Resultados após o reconhecimento	53
7	Conclusões	59
7.1	Trabalhos futuros	60
7.1.1	Classificação das espécies parasitárias	60
7.1.2	Desenvolvimento de software para diagnóstico	61
7.1.3	Avaliação do sistema com mais imagens	61
7.1.4	Automação do sistema	61
	Bibliografia	63

Lista de Tabelas

4.1	Protozoários intestinais mais comuns, doenças que causam, formas e tamanhos.	32
4.2	Helmintos intestinais mais comuns, doenças que causam, formas e tamanhos.	33
6.1	Total de componentes delineados e o número de componentes com erro abaixo de 2.	50
6.2	Erro máximo permitido para identificação das espécies, número de componentes igual ou abaixo do erro máximo e a taxa de acerto do delineamento.	51
6.3	Total de componentes que passam pelos classificadores, valores <i>VP</i> , <i>VN</i> e <i>FP</i> , e percentual de acertos na classificação.	54
6.4	Número de parasitos da base, número de parasitos detectados corretamente (<i>VP</i>), número de não-parasitos detectados incorretamente como parasitos (<i>FP</i>), e percentual de acertos considerando a quantidade <i>VP</i>	55
6.5	Número de acertos e erros para o par de classificadores Rede Neural com <i>Bagging</i> e Rede Neural com <i>Adaboost</i>	55
6.6	Número de acertos e erros para o par de classificadores Rede Neural com <i>Bagging</i> e OPF.	56
6.7	Número de acertos e erros para o par de classificadores Rede Neural com <i>Bagging</i> e SVM.	56
6.8	Número de acertos e erros para o par de classificadores Rede Neural com <i>Bagging</i> e SVM com <i>Adaboost</i>	56
6.9	Número de acertos e erros para o par de classificadores Rede Neural com <i>Adaboost</i> e OPF.	56
6.10	Número de acertos e erros para o par de classificadores Rede Neural com <i>Adaboost</i> e SVM.	56
6.11	Número de acertos e erros para o par de classificadores Rede Neural com <i>Adaboost</i> e SVM com <i>Adaboost</i>	57
6.12	Número de acertos e erros para o par de classificadores OPF e SVM. . . .	57
6.13	Número de acertos e erros para o par de classificadores OPF e SVM com <i>Adaboost</i>	57

6.14	Número de acertos e erros para o par de classificadores SVM e SVM com <i>Adaboost</i>	57
6.15	Valores calculados de χ^2 para os pares de classificadores. Em destaque estão os pares com a hipótese nula rejeitada (os classificadores possuem desempenhos diferentes).	58

Lista de Figuras

2.1	Exemplos de relação de adjacência Euclideana. (a) $\rho = 1$. (b) $\rho = \sqrt{2}$. (c) $\rho = \sqrt{5}$	6
2.2	(a) Os principais elementos de uma floresta. (b) Uma imagem como um grafo orientado com adjacência-4, no qual os números inteiros são valores da imagem $I(t)$ e cada pixel t é representado por um ponto. (c) Uma floresta de caminhos ótimos com a função de custo definida na equação 2.2.	9
3.1	(a) Modelo de um neurônio artificial. (b) Exemplo de camadas de uma rede neural.	15
3.2	Ilustração do hiperplano ótimo em um espaço bidimensional.	19
3.3	Arquitetura de uma máquina de vetores de suporte com m núcleos.	22
3.4	Exemplo de treinamento e teste do classificador Floresta de Caminhos Ótimos. (a) Grafo completo de treinamento com cinco amostras. Os nós A, B e C são da classe ω_1 , e os nós D e E são da classe ω_2 . (b) Árvore geradora mínima executada sobre o grafo completo. (c) Os nós C e D são de classes diferentes e estão ligados por uma aresta na árvore geradora mínima. Portanto, estes nós se tornam protótipos no grafo completo. (d) A partir dos protótipos é calculada uma floresta de caminhos ótimos. Os nós C e D são raízes das árvores de caminhos ótimos. (e) Para a classificação de uma amostra de teste X, o classificador avalia os caminhos dos protótipos até a amostra. A classe atribuída à amostra é a classe do protótipo que oferece o caminho ótimo para esta amostra.	25
3.5	(a) Curva de aprendizado para a base MPEG-7. (b) Curva de aprendizado para a base “meio-anel”. (c) Ilustração da base sintética “meio-anel”.	27
4.1	Sistema com aquisição automática de imagens. As setas normais indicam a seqüência das etapas para a aquisição das imagens, e as setas tracejadas indicam sinais de controle dos equipamentos. O computador controla a bomba peristáltica e os dispositivos para a captura das imagens.	31

4.2	Exemplo de espécies de protozoários mais comuns. (a) <i>Entamoeba histolytica</i> ; (b) <i>Giardia intestinalis</i> ; (c) <i>Entamoeba coli</i> ; (d) <i>Endolimax nana</i> ; (e) <i>Iodamoeba bütschlii</i> ; (f) <i>Blastocystis hominis</i>	33
4.3	Exemplo de espécies de helmintos mais comuns. (a) <i>Enterobius vermiculares</i> ; (b) <i>Trichuris trichiura</i> ; (c) <i>Hymenolepis nana</i> ; (d) <i>Taenia spp.</i> ; (e) <i>Ascaris lumbricoides</i> ; (f) Ancilostomatídeos; (g) <i>Hymenolepis diminuta</i> ; (h) <i>Schistosoma mansoni</i> ; (i) Larva de <i>Strongyloides stercoralis</i>	34
5.1	(a,b) Exemplos de larvas com partes do parasito que se sobrepõem, criando uma região de fundo de imagem no interior. (c) Exemplo de imagem de um ovo de <i>Taenia spp.</i> com impurezas de tamanhos variados; (d) Exemplo de imagem de uma <i>Entamoeba coli</i> com impureza de forma e coloração semelhante a uma <i>Giardia intestinalis</i> ; (e) Exemplo de imagem de ovo de <i>Trichuris trichiura</i> com parte do contorno coberto por impurezas. A imagem apresenta também um ovo de <i>Ascaris lumbricoides</i> na parte inferior; (f) Exemplo de imagem de ovo de <i>Schistosoma mansoni</i> com parte do contorno coberto por impurezas.	37
5.2	Etapas do <i>pipeline</i> para protozoários e ovos de helmintos.	39
5.3	Etapas do <i>pipeline</i> para larvas de helmintos.	39
5.4	Detalhes da etapa de localização de elipses. (a) Etapa para protozoários e ovos de helmintos. (b) Etapa para larvas.	41
5.5	(a) Imagem 5.1e após a etapa de quantização. (b) Imagem a após a etapa de realce de borda. (c) Imagem a após a limiarização e reconstrução superior. (d) Ilustração das elipses encontradas pelo <i>pipeline</i> para ovos, com a imagem original ao fundo. (e) Resultado do <i>watershed</i> para a imagem 5.1a no <i>pipeline</i> para larvas. (f) Resultado da imagem e após a etapa de união de contornos. A seta indica o local de união dos contornos.	44
5.6	(a-f) Exemplos de marcadores para a transformada de <i>watershed</i> , obtidos através das elipses ilustradas na figura 5.5d. A elipse azul representa o marcador interno (componente), e a elipse vermelha representa o marcador externo (fundo da imagem). Na figura (b) o marcador está sobre um ovo de <i>Trichuris trichiura</i> , e na figura (f) o marcador está sobre um ovo de <i>Ascaris lumbricoides</i>	45
5.7	Exemplos de componentes delineados para os marcadores ilustrados na figura 5.6. Os componentes são usados para o reconhecimento. Todos os classificadores rotulam corretamente os componentes (b) e (f) como parasitos, e os outros componentes como impurezas.	45

6.1	(a) Curva de aprendizado para a base de protozoários. (b) Curva de aprendizado para a base de ovos. (c) Curva de aprendizado para a base de larvas.	52
7.1	(a) Exemplo de descritor composto - As medidas de similaridade dos descritores D1 a D4 são combinadas através do módulo de programação genética para se obter uma única medida de saída. (b) Exemplo de árvore de expressões matemáticas gerada por programação genética. As folhas são os valores de similaridade dos descritores.	61

Capítulo 1

Introdução

A segmentação é o processo de particionar a imagem em regiões relevantes para uma dada aplicação. A segmentação constitui etapa fundamental de sistemas de reconhecimento de padrões, sendo responsável pela extração de objetos cujas características são usadas para o reconhecimento. Nesta pesquisa, o método de segmentação desenvolvido é utilizado para a separação automática de parasitos (protozoários e helmintos) do fundo da imagem e impurezas presentes, como detritos, resíduos alimentares, microorganismos, células animais e vegetais.

Poucos são os trabalhos de pesquisa em segmentação e análise de imagens de parasitos intestinais em humanos [45], sendo a maioria sobre as técnicas aplicadas à detecção de parasitos em animais [13, 42]. Esses trabalhos abordam apenas a detecção de ovos, e tratam somente o caso em que os parasitos estão totalmente isolados de outras impurezas nas imagens de amostras fecais.

A pesquisa faz parte de um projeto mais amplo, cujo objetivo geral é o desenvolvimento de um sistema de diagnóstico automático de parasitos intestinais do Homem¹, com meta inicial para o diagnóstico de parasitos de maior prevalência no país. Neste contexto, para uma alta sensibilidade diagnóstica a segmentação deve separar todos os tipos de parasitos, e considerar os casos nos quais os parasitos estão parcialmente encobertos por impurezas.

O projeto principal é multidisciplinar e conta também com a participação de um aluno de doutorado em Biologia, mestre em Parasitologia, e de um aluno de doutorado em Computação. O aluno da Biologia é responsável pela coleta e processamento de amostras fecais, aquisição das imagens e especificação das características dos parasitos, e o aluno da Computação trabalha na pesquisa de classificadores de padrões em imagens.

¹Na biologia, o termo Homem é geralmente utilizado para referir o conjunto de todos os seres humanos.

1.1 Motivação do projeto principal

Segundo a Organização Mundial de Saúde (OMS), atualmente há aproximadamente 2 bilhões de pessoas infectadas com parasitos intestinais. Além de provocar doenças, ou no pior caso levar a óbitos, as parasitoses intestinais podem prejudicar o desenvolvimento físico e intelectual das crianças infectadas. No Brasil, por exemplo, existem aproximadamente 11 milhões de pessoas infectadas por protozoários intestinais e em torno de 125 milhões de pessoas infectadas por helmintos intestinais [38,41]. Apesar do elevado número de pessoas infectadas, a sensibilidade diagnóstica obtida pelos métodos tradicionais é baixa, variando de 48,3% a 75,9% [29]. Os métodos convencionais mais empregados em rotinas laboratoriais são: Lutz (1919) e Hoffman, Pons & Janer (1934); Faust et al. (1939); Rugai, Mattos & Brisola (1954); e Kato-Katz (1960). Esses métodos freqüentemente são processados no campo através de coleta única (frasco coletor universal), em geral sem a utilização de líquido preservador, o que em muitas vezes faz com que estruturas parasitárias cheguem alteradas no laboratório, comprometendo o diagnóstico. No laboratório, cada método apresenta um processamento diferenciado, sendo em geral trabalhoso, este o motivo da demora dos resultados diagnósticos laboratoriais. O método parasitológico empregado para o desenvolvimento do atual projeto (TF-Test) [26] diferencia-se dos demais métodos acima citados, por apresentar: líquido preservador para a coleta; tríplice coleta de material fecal em dias alternados, dia sim, dia não; fácil manuseio para o processamento laboratorial; e totalmente descartável. Essas diferenças facilitam a coleta de campo, favorecem o processamento laboratorial e aumentam a sensibilidade diagnóstica.

Uma avaliação realizada pelo INMETRO², envolvendo vários exames parasitológicos e grandes laboratórios das cidades Rio de Janeiro e São Paulo, e divulgada no programa Fantástico da Rede Globo de Televisão em 14 de setembro de 1997, concluiu como insatisfatória a análise desses laboratórios, pois 3 entre 4 laboratórios de São Paulo apresentaram uma quantidade significativa de erros no diagnóstico, e 7 entre 10 laboratórios do Rio de Janeiro também erraram significativamente. Desta maneira, o diagnóstico automático visa minimizar a taxa de erro. O sistema automático permite a análise de uma quantidade de imagens muito acima da capacidade humana, além de eliminar o uso de vários utensílios para exame.

1.2 Objetivos

A pesquisa tem como objetivo o estudo e desenvolvimento de um método de segmentação automática, com alta sensibilidade (alta taxa de verdadeiros positivos) e especificidade

²Disponível online no endereço <http://www.inmetro.gov.br/consumidor/produtos/laboratorios.asp> em 27/02/2007

(baixa taxa de falsos positivos), para separar os parasitos intestinais do Homem de outras estruturas presentes nas imagens de amostras fecais.

1.3 Contribuições

O método de segmentação automática desenvolvido tem como resultado uma sensibilidade diagnóstica de 97,36% na detecção de parasitos, com 4,73% de falsos positivos. Este resultado é extremamente relevante, considerando a baixa sensibilidade diagnóstica obtida pelos métodos tradicionais. O resultado viabiliza o desenvolvimento de um sistema para diagnóstico, que permite a análise automática das imagens e separação dos parasitos detectados para o diagnóstico final por um especialista. Uma patente que contempla este sistema foi depositada e protocolada sob número 018060127002 no Instituto Nacional da Propriedade Industrial (INPI).

Nós desenvolvemos também um novo classificador de padrões, baseado em floresta de caminhos ótimos, descrito na Seção 3.3. O classificador apresentou desempenho superior aos outros classificadores testados, Rede Neural MLP (Seção 3.1) e Máquina de Vetores de Suporte (Seção 3.2).

Nós elaboramos um artigo sobre o novo classificador [40], com experimentos usando bases de dados públicas, que foi submetido e está em fase de avaliação. Para evitar qualquer problema com o depósito da patente, decidimos não escrever um artigo sobre o método de segmentação de parasitos.

1.4 Organização do trabalho

Os capítulos 2 e 3 apresentam brevemente os fundamentos de processamentos de imagens e reconhecimento de padrões utilizados na pesquisa.

O capítulo 4 descreve o procedimento de aquisição das imagens, e apresenta os parasitos e a base de imagens criada.

O capítulo 5 apresenta o problema de segmentação e descreve o método proposto.

O capítulo 6 apresenta detalhes dos experimentos e os resultados obtidos.

O capítulo 7 apresenta as conclusões da pesquisa e as atividades propostas para a continuidade do trabalho.

Capítulo 2

Fundamentos de Processamento de Imagens

2.1 Imagem Digital

Uma imagem digital \mathbf{I} é um par (\mathcal{I}, \vec{I}) onde \mathcal{I} é um conjunto finito de pixels (pontos em \mathbb{Z}^2), e \vec{I} é um mapeamento vetorial que associa a cada pixel p em \mathcal{I} um conjunto $I_1(p), I_2(p), \dots, I_k(p)$ de valores escalares, associados a alguma propriedade física. O valor de k refere-se ao número de bandas.

2.1.1 Imagem em tons de cinza

Uma imagem \mathbf{I} em tons de cinza (e.g., foto, imagem de ultra-som) é uma matriz de tamanho $N \times M$ pixels (N linhas e M colunas), onde \vec{I} possui apenas uma banda ($k = 1$). Neste caso, será usada a notação I no lugar de \vec{I} . O valor $I(p)$ de cada pixel p é obtido por amostragem e quantização de uma função contínua $I_c(x, y)$ que descreve a propriedade física correspondente em uma dada região do espaço. No caso de uma foto temos o brilho, e no caso de uma tomografia de raios-X temos a densidade do tecido. Normalmente, valores altos são apresentados na tela como pixels claros e valores baixos como pixels escuros.

2.1.2 Imagem binária

Uma imagem \mathbf{I} binária é uma matriz de tamanho $N \times M$ pixels, com somente dois valores possíveis $I(p)$ para cada pixel p . Geralmente as imagens binárias são usadas como máscaras, onde os pixels de valor $I(p)$ alto representam pontos de interesse e pixels de valor baixo representam pontos sem relevância.

2.1.3 Imagem multibanda

Uma imagem \mathbf{I} é multibanda quando $|\vec{I}| = k > 1$. Um exemplo de imagem multibanda é uma foto colorida RGB, com $k = 3$, onde as bandas correspondem às cores vermelha, verde e azul.

As imagens de parasitos usadas neste trabalho são RGB, e as imagens em tons de cinza e binárias são obtidas através de transformações da imagem multibanda.

2.1.4 Amostragem e quantização

Cada pixel é amostrado a intervalos (Δ_x, Δ_y) (e.g., $\Delta_x = \Delta_y = 1\text{mm}$). Quanto menor for o intervalo de amostragem para uma mesma região do espaço, maior será a resolução espacial da imagem. As imagens de parasitos possuem 2560 colunas e 1920 linhas de pixels, e cobrem uma região de $300\mu\text{m}$ de comprimento por $225\mu\text{m}$ de largura, com intervalo de amostragem $\Delta_x = \Delta_y = 0,1172\mu\text{m}$.

Os valores de $I_c(x, y)$ amostrados são quantizados em 2^b níveis de cinza, onde b é chamado profundidade da imagem em bits (e.g., $b = 8$, profundidade de 8 bits). Quanto maior o intervalo de quantização, menor é a resolução radiométrica da imagem (i.e., um intervalo grande de valores de $I_c(x, y)$ será mapeado para o mesmo valor quantizado).

2.2 Topologia Digital

2.2.1 Relação Binária

Uma relação binária R aplicada a um conjunto X é um subconjunto do produto cartesiano $X \times X$. Uma relação binária é dita reflexiva se $(a, a) \in R$, para todo $a \in X$, simétrica se $(a, b), (b, a) \in R$, para todo $a, b \in X$, e transitiva se $(a, b), (b, c) \in R$ implica que $(a, c) \in R$, para todo $a, b, c \in X$.

2.2.2 Métrica

Uma função d de distância entre pixels é uma métrica se:

$$\begin{aligned} d(p, q) &\geq 0, \\ d(p, q) &= d(q, p), \\ d(p, r) &\leq d(p, q) + d(q, r) \end{aligned}$$

onde $p = (x_p, y_p)$, $q = (x_q, y_q)$ e $r = (x_r, y_r)$ são três pixels da imagem. Como exemplos podemos citar a métrica Euclideana, onde $d(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$ e a métrica City-block, onde $d(p, q) = |x_p - x_q| + |y_p - y_q|$.

2.2.3 Relação de Adjacência e Grafos

Uma relação de adjacência \mathcal{A} é uma relação binária entre pixels de \mathcal{I} . Uma vez definida a relação \mathcal{A} , a imagem \mathbf{I} pode ser interpretada como um grafo orientado cujos nós são os pixels e os arcos são pares de pixels em \mathcal{A} . Na maioria das aplicações, a relação \mathcal{A} é invariante à translação, o que significa que depende apenas da posição relativa dos pixels. Por exemplo, \mathcal{A} pode ser definida como consistindo de todos os pares de pixels distintos $(p, q) \in \mathcal{I} \times \mathcal{I}$ tal que $d(p, q) \leq \rho$, onde $d(p, q)$ denota a distância Euclideana e ρ é uma constante especificada. As Figuras 2.1a-c mostram os pixels adjacentes de um pixel central s para $\rho = 1$ (vizinhos-4), $\rho = \sqrt{2}$ (vizinhos-8), e $\rho = \sqrt{5}$, respectivamente.

No grafo definido por \mathcal{A} , um caminho π é uma seqüência de pixels adjacentes $\langle p_1, p_2, \dots, p_k \rangle$, onde $(p_i, p_{i+1}) \in \mathcal{A}$, $i = 1, 2, \dots, k - 1$. O caminho π é dito trivial se $\pi = \langle p_1 \rangle$.

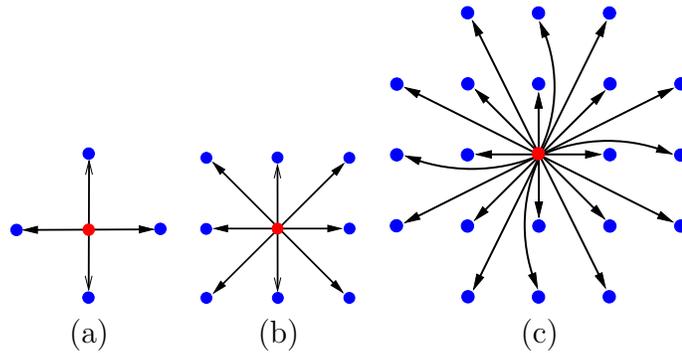


Figura 2.1: Exemplos de relação de adjacência Euclideana. (a) $\rho = 1$. (b) $\rho = \sqrt{2}$. (c) $\rho = \sqrt{5}$.

2.2.4 Relação de Conexidade

Um pixel q é conexo a um pixel p se existir um caminho de p a q no grafo definido por \mathcal{A} . Dizemos que dois pixels são conexos-4 se forem ligados por caminhos cujos pixels adjacentes são vizinhos-4.

2.2.5 Componente Conexa

Um componente conexo na imagem \mathbf{I} é um subconjunto de \mathcal{I} , onde todos os pares (p, q) de pixels são conexos (i.e., existe um caminho de p a q e um caminho de q a p , não necessariamente o mesmo caminho).

2.3 Transformada Imagem-Floresta

A Transformada Imagem-Floresta (IFT) [21] é uma ferramenta genérica para o projeto, implementação e avaliação de operadores de imagem baseados em conexidade. A IFT define uma floresta de caminhos mínimos em um grafo, cujos nós são os pixels da imagem e as arestas são definidas por uma relação de adjacência entre pixels. O custo de um caminho no grafo é determinado por uma função de custo específica para cada aplicação, e normalmente depende de propriedades locais da imagem ao longo do caminho, como cor, gradiente e posição do pixel. A IFT associa a cada pixel da imagem um caminho de custo mínimo, particionando a imagem em uma floresta de caminhos ótimos onde cada árvore tem como raiz um pixel semente e como nós os pixels da imagem mais “conexos” com a raiz do que com qualquer outra semente, em algum sentido apropriado. A IFT retorna três atributos para cada pixel: seu predecessor no caminho ótimo, o custo do caminho e sua raiz correspondente (ou um rótulo associado).

A IFT unifica e estende muitas técnicas de análise de imagem que, apesar de serem baseadas em conceitos semelhantes (propagação ordenada, inundação, dilatação geodésica, programação dinâmica, crescimento de regiões, busca A^* , etc.), são apresentadas como métodos não relacionados. Tais técnicas podem ser reduzidas a uma partição da imagem em zonas de influência associadas a um conjunto de sementes, onde a zona de cada semente consiste de pixels que são mais conexos a essa semente do que a qualquer outra. As zonas de influência são as árvores da floresta definida pela IFT. A IFT também possui suporte para muitos operadores de processamento de imagem que não são claramente relacionados com partição da imagem, como reconstruções morfológicas [20], transformadas de distância [19, 32], esqueletos multi-escala [19], saliências e dimensão fractal multi-escala [12] e perseguição de contornos [22–24].

Neste trabalho, a IFT é aplicada em imagens em tons de cinza e imagens binárias.

2.3.1 Definições

Um caminho π é uma seqüência de pixels adjacentes $\langle t_1, t_2, \dots, t_k \rangle$, $(t_i, t_{i+1}) \in \mathcal{A}$, $i = 1, 2, \dots, k-1$, e um caminho π trivial, com somente um pixel, é representado por $\pi = \langle t_1 \rangle$.

Seja π um caminho que termina em um pixel t , e τ um caminho que inicia em t . $\pi \cdot \tau$ é o caminho resultante da concatenação de π e τ , com as duas instâncias de t unidas em uma.

O custo de um caminho π é determinado por uma função de custo de caminho $f(\pi)$ adequada para o problema, a qual normalmente depende de propriedades locais da imagem ao longo do caminho - tais como cor, gradiente, e posição de pixel. Exemplos de funções

de custo de caminho são:

$$h(t_1) + \sum_{i=1}^{k-1} w(t_i, t_{i+1}), \quad (2.1)$$

$$\max_{i=1,2,\dots,k-1} \{h(t_1), w(t_i, t_{i+1})\}, \quad (2.2)$$

$$h(t_1) + \left(\sum_{i=1}^{k-1} |x_{t_i} - x_{t_{i+1}}| \right)^2 + \left(\sum_{i=1}^{k-1} |y_{t_i} - y_{t_{i+1}}| \right)^2, \quad (2.3)$$

onde $h(t_1)$ é o custo $f(\langle t_1 \rangle)$ do caminho trivial $\langle t_1 \rangle$, $w(t_i, t_{i+1})$ é um peso fixo e não negativo associado ao arco $(t_i, t_{i+1}) \in \mathcal{A}$, e (x_{t_i}, y_{t_i}) são as coordenadas em \mathbb{Z}^2 do pixel t_i .

Um caminho π é dito ótimo se $f(\pi) \leq f(\pi')$ para qualquer outro caminho π' que termine no mesmo pixel t independente do pixel inicial. Neste caso, o valor de $f(\pi)$ é por definição o custo do pixel t .

Um mapa de predecessores é uma função P que associa a cada pixel t em \mathcal{I} ou algum outro pixel em \mathcal{I} , ou um marcador *nil* que não pertence a \mathcal{I} — neste caso t é dito ser uma raiz do mapa — isto é, aquela função que leva todos os pixels a *nil* em um número finito de iterações, formando uma floresta de pixels. Para qualquer pixel $t \in \mathcal{I}$, uma floresta P define um caminho $\pi^*(t)$ recursivamente como $\langle t \rangle$ se $P(t) = \text{nil}$, e $\pi^*(s) \cdot \langle s, t \rangle$ se $P(t) = s \neq \text{nil}$. $P^0(t)$ é o pixel inicial ou raiz de $\pi^*(t)$ (Ver figura 2.2a).

2.3.2 IFT

Uma IFT é, portanto, uma transformação que, para uma dada imagem \mathbf{I} , relação de adjacência \mathcal{A} e função de custo de caminho f , calcula uma floresta de caminhos ótimos. O resultado da IFT é uma imagem anotada que associa a cada pixel três atributos: seu predecessor no caminho ótimo, o custo deste caminho, e o pixel raiz correspondente. Uma alternativa usada em algumas aplicações é associar um rótulo para cada raiz, tal que o mapa de raízes é substituído por um mapa de rótulos. Uma função de rotulação $\lambda(t)$ é definida para retornar o pixel raiz de t em aplicações que usam mapa de raízes, e o rótulo do pixel t em aplicações que usam mapa de rótulos.

Em muitas aplicações, existe ainda um conjunto especial $\mathcal{S} \subseteq \mathcal{I}$ de pixels chamados sementes. Este conjunto exerce influência direta no cálculo da função de custo de um caminho, pois define um mapa de custos inicial para o problema. Isto é, ele define um valor finito para $h(t_1)$, se $t_1 \in \mathcal{S}$, ou $+\infty$ no caso contrário. Um pixel t_k em \mathcal{S} é na verdade um candidato a raiz da floresta, que só será raiz se não existir nenhum outro pixel t_1 em \mathcal{S} tal que $f(\langle t_1, t_2, \dots, t_k \rangle) < f(\langle t_k \rangle)$.

A Figura 2.2b ilustra um exemplo de um grafo orientado com adjacência-4, no qual os números inteiros representam os valores de imagem $I(t)$ e cada pixel t é representado

por um ponto. Considere a função de custo de caminho definida na equação 2.2, onde $w(t_i, t_{i+1}) = I(t_{i+1})$, $h(t_1) = I(t_1)$ para os pixels sementes representados por pontos com contorno em vermelho, e $h(t_1) = +\infty$ para os outros pixels. Uma floresta de caminhos ótimos para este caso é apresentada na Figura 2.2c.

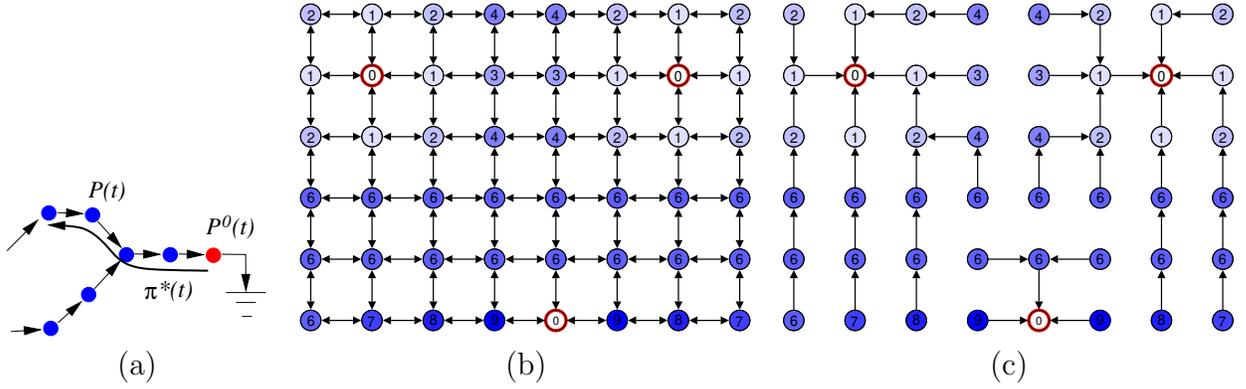


Figura 2.2: (a) Os principais elementos de uma floresta. (b) Uma imagem como um grafo orientado com adjacência-4, no qual os números inteiros são valores da imagem $I(t)$ e cada pixel t é representado por um ponto. (c) Uma floresta de caminhos ótimos com a função de custo definida na equação 2.2.

A IFT é calculada por um variante do algoritmo de Dijkstra/Moore/Dial [15, 16, 36] e pode ser obtida em tempo linear com o número de pixels na maioria das aplicações [21]. O algoritmo geral da IFT é apresentado abaixo:

Algoritmo 1. ALGORITMO GERAL DA IFT

ENTRADA: Imagem $\mathbf{I} = (\mathcal{I}, I)$, relação de adjacência \mathcal{A} , função de rotulação λ e função de custo de caminho f .
 SAÍDA: Imagem de custo \mathbf{C} , imagem de predecessores \mathbf{P} , e imagem de raízes \mathbf{R} .
 AUXILIARES: Fila de prioridade Q e variável cst .

1. **Para cada** $t \in \mathcal{I}$ **faça**
2. $C(t) = f(\langle t \rangle)$, $P(t) = nil$, e $R(t) = \lambda(t)$.
3. **Se** $C(t) < +\infty$ **então** *Insira* t *em* Q .
4. **Enquanto** $Q \neq \emptyset$ **faça**
5. *Remova um pixel* s *de* Q *cujos custo* $C(s)$ *é mínimo.*
6. **Para cada** t *tal que* $(s, t) \in \mathcal{A}$ **faça**
7. **Se** $C(t) > C(s)$ **então**
8. $cst = f(\pi^*(s) \cdot \langle s, t \rangle)$.
9. **Se** $cst < C(t)$ **então**
10. **Se** $C(t) \neq +\infty$ **então** *Remova* t *de* Q .
11. $C(t) = cst$, $P(t) = s$, $R(t) = R(s)$, e *insira* t *em* Q .

Ao final do algoritmo as imagens de saída estarão com os atributos de custo, pixel predecessor e raiz (ou rótulo) de cada pixel.

2.3.3 Aplicações

Muitos operadores podem ser reduzidos a uma operação local simples sobre a imagem anotada. Alguns exemplos são apresentados abaixo:

- Transformada de *watershed* por marcadores [20, 31]: \mathcal{A} é arbitrária, \mathcal{S} são os pixels marcadores, e f é dada pela equação 2.2, onde $w(t_i, t_{i+1}) = I(t_{i+1})$ e $h(t_1) = I(t_1)$, se $t_1 \in \mathcal{S}$, ou $+\infty$ no caso contrário. A partição de *watershed* é obtida diretamente do mapa de rótulos.
- Reconstrução superior por marcadores [20]: \mathcal{A} é arbitrária, \mathcal{S} são os pixels marcadores, J é uma imagem onde $J(t) > I(t)$ se $t \in \mathcal{S}$, e f é dada pela equação 2.2, onde $w(t_i, t_{i+1}) = I(t_{i+1})$ e $h(t_1) = J(t_1)$, se $t_1 \in \mathcal{S}$, ou $+\infty$ no caso contrário. A imagem reconstruída é obtida diretamente do mapa de custos e uma família de transformadas de *watershed* é obtida do mapa de rótulos para diferentes valores de $J(t_1) > I(t_1)$. Neste caso, a IFT mostra claramente a relação entre reconstrução morfológica e transformada de *watershed*.
- Perseguição de bordas [22–24]: \mathcal{A} é vizinhos-4 ou -8, \mathcal{S} é um dado pixel sobre a borda, e f é dada pela equação 2.1, onde $w(t_i, t_{i+1}) = |I(t_{i+1}) - I(t_i)|$, por exemplo, e $h(t_1) = 0$, se $t_1 \in \mathcal{S}$, ou $+\infty$ no caso contrário. O caminho ótimo de \mathcal{S} a um dado pixel t da borda, que forma um segmento ótimo de borda, é obtido diretamente do mapa de predecessores $P(t)$.
- Transformada de distância Euclideana e operadores relacionados [11, 19]: \mathcal{A} é vizinhos-8 ou maior, \mathcal{S} são pixels de contorno, e f é dada pela equação 2.3, onde $h(t_1) = 0$, se $t_1 \in \mathcal{S}$, ou $+\infty$ no caso contrário. Os valores de distância são obtidos no mapa de custos e o mapa de rótulos pode ser usado para obter esqueletos multi-escala [19], com um pixel de largura e conexos em todas as escalas. Os mapas de custos e rótulos também podem ser usados para calcular a dimensão fractal multi-escala e os pontos de saliência da forma [11]. Para o caso da saliência, cada pixel do contorno conquista um determinado número de pixels ao redor, definindo uma área de influência na parte interna e externa do contorno. O pixel é classificado como convexo quando a área de influência externa é maior que a interna, e é classificado como côncavo no caso contrário.
- Caminho Geodésico entre pixels s e t : \mathcal{A} é vizinhos-8, $\mathcal{S} = \{s\}$ e f é dada pela equação 2.1, onde $h(t_1) = 0$, se $t_1 = s$, ou $+\infty$ no caso contrário. O custo do

caminho pode ser calculado pela métrica chanfrada (*chamfer*), com $w(t_i, t_{i+1}) = a \max\{|x_{t_i} - x_{t_{i+1}}|, |y_{t_i} - y_{t_{i+1}}|\} + (b - a) \min\{|x_{t_i} - x_{t_{i+1}}|, |y_{t_i} - y_{t_{i+1}}|\}$, onde a e b são constantes com $0 < a \leq b \leq 2a$. O caminho geodésico de s a t é obtido do mapa de predecessores $P(t)$.

- Mínimos regionais: \mathcal{A} é arbitrária, \mathcal{S} são todos os pixels da imagem, e f é definida como

$$f(\langle t \rangle) = I(t), \text{ para todo } t \in \mathcal{I},$$

$$f(\pi \cdot \langle s, t \rangle) = \begin{cases} f(\pi), & \text{se } I(s) \leq I(t); \\ +\infty, & \text{caso contrário.} \end{cases}$$

Inicialmente, todos os pixels são raízes ($P(t) = nil$). Os pixels de menor custo conquistam pixels adjacentes de custo maior, gerando árvores enraizadas nesses pixels de menor custo. Ao final do processamento da IFT, os mínimos regionais são todos os pixels que continuam como raízes.

- Árvore geradora mínima: \mathcal{A} é arbitrária, \mathcal{S} é algum pixel inicial. O algoritmo da IFT pode ser utilizado para o cálculo de uma árvore geradora mínima, com o uso da função de custo

$$f(\pi \cdot \langle s, t \rangle) = D(s, t)$$

onde $D(s, t)$ é o peso da aresta entre s e t , ou seja, o custo de um caminho é o peso da última aresta no caminho. Com isso, o algoritmo da IFT se transforma no algoritmo de Prim [8], e a árvore geradora mínima é obtida do mapa de predecessores. A árvore orientada definida no mapa de predecessores é processada e as orientações das arestas são removidas, resultando na árvore geradora mínima.

2.4 Realce de bordas

Uma imagem com as bordas dos objetos realçadas pode ser obtida através do cálculo do gradiente da imagem, de forma a deixar regiões planas com valores baixos e contornos de objetos com valores altos. Na implementação realizada, a diferença $D(p, q)$ entre dois pixels p e q é dada por

$$D(p, q) = \sum_{\forall r \in \mathcal{A}_p, \forall s \in \mathcal{A}_q} [I(r) - I(s)]$$

e o valor de gradiente $G(p)$ de um pixel p é dado por

$$G(p) = \left\| \sum_{\forall q \in \mathcal{A}_p} \left[\frac{\vec{p}q}{\|\vec{p}q\|} D(p, q) \right] \right\| \quad (2.4)$$

onde \mathcal{A}_p é a relação de adjacência vizinhos-8 do pixel p , \mathcal{A}_q é a relação de adjacência vizinhos-8 do pixel q , e $\frac{\vec{pq}}{\|\vec{pq}\|}$ é o vetor unitário do pixel p para o pixel q .

O uso deste gradiente melhorou os resultados em relação aos gradientes de Sobel e Prewitt [27].

Capítulo 3

Fundamentos de Reconhecimento de Padrões

As técnicas de reconhecimento de padrões são usadas para a classificação de objetos. Os objetos são descritos por um conjunto de medidas, também conhecidas como atributos ou características, que podem ser quantitativas (e.g., área, comprimento, pressão, número de habitantes) ou qualitativas (e.g., nível de instrução, profissão, tipo sanguíneo). O reconhecimento de padrões estatísticos trabalha com características numéricas, e os valores das características de um objeto são colocadas em um vetor $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, onde n é o número de características. O espaço real \mathbb{R}^n é chamado de espaço de características, e cada eixo corresponde a uma característica.

Os problemas de reconhecimento de padrões podem ser divididos em não-supervisionados e supervisionados [30]. No modelo não-supervisionado, não há conhecimento das classes dos dados, e o problema é descobrir a estrutura do conjunto de dados. Por exemplo, se há grupos de dados e quais características são relevantes para a separação dos objetos. No modelo supervisionado, cada objeto tem uma classe definida, e o problema é treinar o classificador de forma a indicar a classe correta para um novo objeto de teste. O modelo supervisionado foi usado na pesquisa, e cada objeto do conjunto de dados foi rotulado como parasito ou não-parasito.

As classes são definidas como $\Omega = \{\omega_1, \dots, \omega_c\}$, onde c é o número de classes, e cada objeto pertence somente a uma classe. O conjunto de dados é representado por $\mathbf{Z} = \{z_1, \dots, z_N\}$, onde N é o número de objetos. O conjunto de dados geralmente é separado em dois conjuntos: treinamento (\mathbf{Z}_{tr}) e teste (\mathbf{Z}_{te}). Quando há uma grande quantidade de amostras disponíveis, pode ser criado também um conjunto de avaliação (\mathbf{Z}_{av}). O conjunto de treinamento é usado para treinar a separabilidade de classes dos classificadores, e o conjunto de avaliação é usado para verificar a corretude do treinamento. Na tentativa de melhorar a taxa de acerto do classificador, algumas amostras podem ser trocadas entre o

conjunto de treinamento e avaliação, e um novo treinamento pode ser realizado. Finalizado o treinamento do classificador, a avaliação da generalização, i.e., a capacidade de classificar corretamente objetos diferentes das amostras usadas no treinamento, é realizada com o conjunto de teste. A classe de uma amostra $z_j \in \mathbf{Z}, j = 1, \dots, N$ é dada por $l(z_j) \in \Omega$.

As seções 3.1, 3.2 e 3.3 descrevem os classificadores Rede Neural, Máquina de Vetores de Suporte e Floresta de Caminhos Ótimos, respectivamente. As seções 3.4.1 e 3.4.2 descrevem as técnicas para coleção de classificadores *Bagging* e *Adaboost*.

3.1 Classificador Rede Neural

O estudo de redes neurais artificiais foi inspirado pela observação de que o cérebro é composto por uma rede complexa de neurônios interconectados. De forma simplificada, as redes neurais artificiais são compostas de várias unidades simples (neurônios ou nós) interconectadas, onde cada unidade recebe um certo número de valores de entrada (possivelmente as saídas de outras unidades), e produz um valor de saída (que pode ser a entrada de várias outras unidades). Há várias arquiteturas de redes neurais presentes na literatura, e vamos apresentar brevemente o modelo MLP (Perceptron Multicamadas) [17, 30, 35], um dos modelos mais populares de redes neurais artificiais e o modelo adotado na pesquisa.

A rede neural recebe um vetor de características \mathbf{x} na entrada, e produz valores para c funções discriminantes $g_1(\mathbf{x}), \dots, g_c(\mathbf{x})$ na saída. Normalmente, o treinamento do classificador é realizado com o objetivo de minimizar o erro quadrático E da classificação de amostras do conjunto de treinamento \mathbf{Z}_{tr} , definido por

$$E = \frac{1}{2} \sum_{j=1}^{N_{tr}} \sum_{i=1}^c [g_i(\mathbf{x}_j) - M(\omega_i, l(z_j))]^2 \quad (3.1)$$

onde N_{tr} é o tamanho do conjunto \mathbf{Z}_{tr} , \mathbf{x}_j é o vetor de características do objeto $z_j \in \mathbf{Z}_{tr}$, e $M(\omega_i, l(z_j))$ é uma função que gera o valor 1 se a classe de z_j for ω_i , ou o valor 0 caso contrário.

A figura 3.1a ilustra o modelo de um neurônio artificial. Seja $\mathbf{u} = [u_0, \dots, u_q]^T \in \mathbb{R}^{q+1}$ o vetor de entrada do neurônio com q conexões, e a saída $v = \phi(\xi) \in \mathbb{R}$ o resultado de uma função de ativação aplicada a $\xi = \sum_{i=0}^q w_i u_i = \mathbf{w}^T \mathbf{u}$, onde $\mathbf{w} = [w_0, \dots, w_q]^T \in \mathbb{R}^{q+1}$ é um vetor de pesos sinápticos. A interpretação geométrica de ξ é de um hiperplano em \mathbb{R}^q . A entrada fixa u_0 , juntamente ao peso w_0 associado, tem o efeito de transladar o hiperplano em torno da origem, para que a saída v do nó não seja nula quando todas as demais entradas $\{u_1, \dots, u_q\}$ forem nulas.

O perceptron é um classificador para duas classes com um único neurônio, e usa a

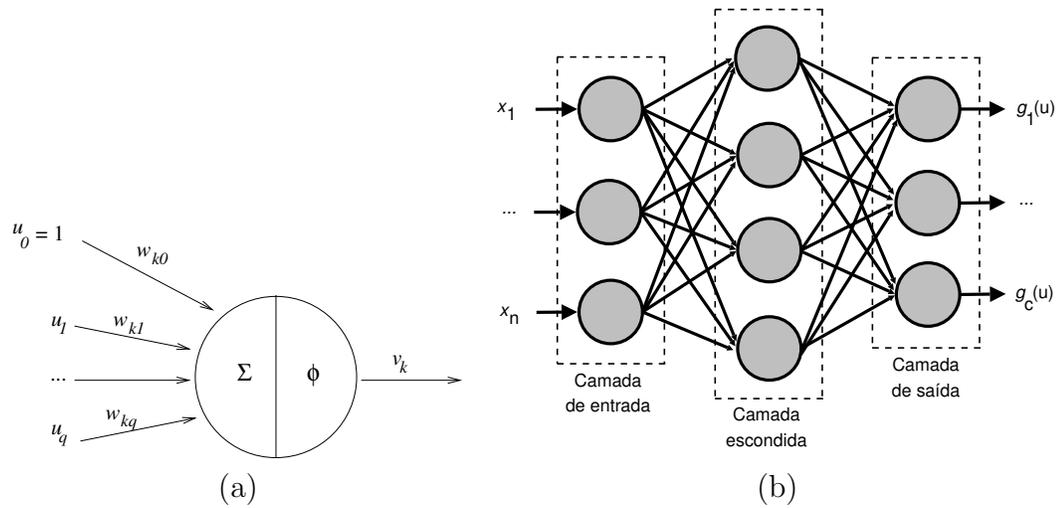


Figura 3.1: (a) Modelo de um neurônio artificial. (b) Exemplo de camadas de uma rede neural.

função de ativação por limiar

$$\phi(\xi) = \begin{cases} 1, & \text{se } \xi \geq 0; \\ -1, & \text{no caso contrário.} \end{cases}$$

Um hiperplano separa o espaço em duas regiões, e cada região representa uma classe. Os pesos sinápticos são encontrados através de um algoritmo de treinamento, que inicialmente atribui pesos aleatórios a \mathbf{w} . Os vetores de características \mathbf{x} das amostras do conjunto \mathbf{Z}_{tr} são colocadas na entrada do perceptron, e o algoritmo modifica os pesos para amostras classificadas incorretamente (aparecem do lado errado do hiperplano). Os pesos sinápticos são corrigidos através da equação

$$\mathbf{w} = \mathbf{w} - v\eta\mathbf{x}_j \quad (3.2)$$

onde v é a saída do perceptron para o vetor de características \mathbf{x}_j e η é um parâmetro que especifica a taxa de aprendizagem. Apesar da simplicidade do algoritmo, o treinamento possui as seguintes propriedades:

- Se duas classes são linearmente separáveis em \mathbb{R}^n , o algoritmo sempre converge em um número finito de passos.
- Se duas classes não são linearmente separáveis em \mathbb{R}^n , o algoritmo ficará em *loop* infinito e nunca irá convergir.

A rede MLP é um conjunto de perceptrons conectados e organizados em camadas. A primeira camada, que recebe o vetor de características, é chamada de camada de entrada,

e a última camada é chamada de camada de saída, ilustrada na figura 3.1b. Entre estas duas camadas estão as camadas escondidas, sem limitações no número de perceptrons, e todos os nós das camadas escondidas possuem a mesma função de ativação ϕ . Os perceptrons estão conectados de forma que a saída de uma camada esteja conectada somente a perceptrons da camada seguinte. Os perceptrons da camada de entrada são nós especiais, cujo papel é exclusivamente distribuir cada uma das entradas da rede a todos os perceptrons da camada seguinte, sem modificar os dados de entrada.

O algoritmo para treinamento supervisionado de redes MLP mais popular é o treinamento por retropropagação, e necessita que a função de ativação seja diferenciável. Neste caso, a função de ativação ϕ mais usada é uma sigmóide, tipicamente definida como

$$\phi(\xi) = \frac{1}{1 + \exp(-\xi)}, \quad \text{e sua derivada} \quad \phi(\xi)' = \phi(\xi)[1 - \phi(\xi)]. \quad (3.3)$$

Basicamente, o algoritmo consiste em duas etapas de processamento: o processamento direto e o processamento reverso. No processamento direto, um vetor de características é aplicado à entrada da rede neural, e os cálculos são propagados pela rede até a camada de saída. Durante este processamento, os pesos dos nós permanecem fixos. No processamento reverso, o erro calculado na saída é propagado no sentido inverso, camada a camada, e ao final do processo os pesos são atualizados de acordo com uma regra de correção de erro. Seja θ um parâmetro da rede neural e $J(\theta)$ uma função de erro a ser minimizada. O método de descida de gradiente (*gradient descent*) atualiza θ por

$$\theta = \theta - \eta \frac{\partial J}{\partial \theta}$$

onde $\eta > 0$ é a taxa de aprendizagem. Sendo $J = E$ e θ um peso w_j , $0 \leq j \leq q$, da j -ésima conexão de entrada de um dado nó da rede com q conexões, $\frac{\partial E}{\partial w_j}$ pode ser obtido por

$$\frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial \xi} \frac{\partial \xi}{\partial w_j} = \frac{\partial E}{\partial \xi} u_j = \delta u_j$$

onde δ é um erro associado ao nó e u_j é sua j -ésima entrada. Portanto, o peso w_j de qualquer conexão de entrada de qualquer nó da rede deve ser atualizado por

$$w_j \leftarrow w_j - \eta \delta u_j. \quad (3.4)$$

Os valores u_j de entrada de todas as conexões de todos os nós são calculados por processamento direto de um vetor de características $\mathbf{x} \in \mathbb{R}^n$ de uma amostra de treinamento $z \in \mathbf{Z}_{tr}$, com classe conhecida $l(z)$. Inicialmente, são atribuídos valores aleatórios pequenos a todos os pesos, e um valor é escolhido para a taxa de aprendizagem η . Após o

processamento direto, o cálculo de δ é realizado da camada de saída para a camada de entrada. A derivada de E em relação à i -ésima saída $g_i(\mathbf{x})$ é

$$\frac{\partial E}{\partial g_i(\mathbf{x})} = g_i(\mathbf{x}) - M(l(z), \omega_i).$$

Seja $g_i(\mathbf{x}) = \phi(\xi_i^o)$ a saída de um nó i na camada de saída o . O erro δ_i^o é encontrado pela regra da cadeia

$$\delta_i^o = \frac{\partial E}{\partial \xi_i^o} = \frac{\partial E}{\partial g_i(\mathbf{x})} \frac{\partial g_i(\mathbf{x})}{\partial \xi_i^o} = [g_i(\mathbf{x}) - M(l(z), \omega_i)] \frac{\partial \phi(\xi_i^o)}{\partial \xi_i^o}.$$

Para a função de ativação ϕ sigmóide da equação 3.3,

$$\delta_i^o = [g_i(\mathbf{x}) - M(l(z), \omega_i)] g_i(\mathbf{x}) [1 - g_i(\mathbf{x})]$$

Portanto, δ_i^o , $i = 1, \dots, c$ são os erros dos c nós na camada de saída e são usados para a atualização dos pesos da última camada escondida para a camada de saída. Para a atualização destes pesos, seja w_{ij}^o o peso da conexão entre o j -ésimo nó na última camada escondida h e o i -ésimo nó na camada de saída o , e

$$w_{ij}^o = w_{ij}^o - \eta \delta_i^o v_j^h$$

Vamos considerar agora as conexões entre a penúltima camada (camada escondida ou de entrada) $h - 1$ e a última camada escondida h . A saída v_k^{h-1} do k -ésimo nó da camada $h - 1$ é a entrada u_k do j -ésimo nó da camada h , a qual é multiplicada pelo peso w_{jk}^h . A saída do j -ésimo nó é v_j^h . Pela equação 3.4, a atualização dos pesos é dada por

$$w_{jk}^h = w_{jk}^h - \eta \delta_j^h v_k^{h-1}$$

Temos v_k^{h-1} por processamento direto de \mathbf{x} , e o erro δ_j^h de cada j -ésimo nó da última camada escondida é encontrado através da equação

$$\delta_j^h = \frac{\partial E}{\partial \xi_j^h} = \frac{\partial E}{\partial v_j^h} \frac{\partial v_j^h}{\partial \xi_j^h} = \frac{\partial E}{\partial v_j^h} \frac{\partial \phi(\xi_j^h)}{\partial \xi_j^h} \quad (3.5)$$

Para a função de ativação ϕ sigmóide temos

$$\delta_j^h = \frac{\partial E}{\partial v_j^h} v_j^h (1 - v_j^h) \quad (3.6)$$

A saída v_j^h afeta o erro E através de todos os nós da camada de saída o . Então,

$$\frac{\partial E}{\partial v_j^h} = \sum_{i=1}^c \frac{\partial E}{\partial \xi_i^o} \frac{\partial \xi_i^o}{\partial v_j^h} = \sum_{i=1}^c \delta_i^o w_{ij}^o \quad (3.7)$$

Os erros δ_i^o já foram calculados para os nós da camada de saída. Usando as equações 3.6 e 3.7, o erro do j -ésimo nó é dado por

$$\delta_j^h = \left(\sum_{i=1}^c \delta_i^o w_{ij}^o \right) v_j^h (1 - v_j^h) \quad (3.8)$$

O procedimento é repetido para todos os nós das camadas anteriores.

Há duas formas de treinamento para um conjunto de amostras \mathbf{Z}_{tr} : o treinamento por lote, onde a atualização dos pesos ocorre somente após o processamento de todo o conjunto \mathbf{Z}_{tr} , chamado época, e o treinamento *on-line*, onde a atualização dos pesos ocorre após cada amostra $z_j \in \mathbf{Z}_{tr}$ processada. No treinamento por lote, as atualizações dos pesos sinápticos são somadas durante o processamento direto das amostras, e a atualização ocorre ao final da época. O treinamento é executado por um determinado número de épocas ou até o erro ficar abaixo de um limiar especificado.

3.2 Classificador Máquina de Vetores de Suporte

As máquinas de vetores de suporte (SVM, do inglês *Support Vector Machines*) [5, 28] se baseiam em um mapeamento não-linear $\varphi(\cdot)$ das amostras de entrada para uma dimensão mais alta, de forma que dados de duas classes diferentes sejam separáveis por um hiperplano. O teorema de Cover define que amostras não separáveis em um espaço podem ser mapeadas em um espaço de maior dimensionalidade através de transformações não-lineares, e assim se tornarem linearmente separáveis com alta probabilidade [9].

Considerando um vetor de características \mathbf{x} de amostras do conjunto de treinamento \mathbf{Z}_{tr} , a família $\{y_j = \varphi(\mathbf{x}_i)\}_{j=1}^m, i = 1, \dots, N_{tr}$, de transformações não-lineares sobre \mathbf{Z}_{tr} gera amostras $\mathbf{y}_i = [y_1, \dots, y_m]^T$ no espaço oculto, onde N_{tr} é o tamanho do conjunto \mathbf{Z}_{tr} e m é o número de transformações não-lineares. Sendo \mathbf{w} um vetor de pesos, o problema consiste em encontrar o hiperplano ótimo $\mathbf{w}^{*T} \mathbf{y} + w_0^* = 0$ que separa as amostras o máximo possível, ou seja, uma máquina de vetores de suporte busca os pesos que maximizam as distâncias $|r|$ entre as amostras e o hiperplano de separação. Sendo $g(\mathbf{y}) = \mathbf{w}^{*T} \mathbf{y} + w_0^*$ uma função discriminante que calcula a distância de \mathbf{y} ao hiperplano ótimo, e \mathbf{y}_p a projeção

de um ponto \mathbf{y} sobre o hiperplano, temos

$$\begin{aligned}\mathbf{y} &= \mathbf{y}_p + r \frac{\mathbf{w}^*}{\|\mathbf{w}^*\|} \\ g(\mathbf{y}_p) &= \mathbf{w}^{*T} \mathbf{y}_p + w_0^* = 0 \\ g(\mathbf{y}) &= \mathbf{w}^{*T} \left(\mathbf{y}_p + r \frac{\mathbf{w}^*}{\|\mathbf{w}^*\|} \right) + w_0^* \\ g(\mathbf{y}) &= \mathbf{w}^{*T} \mathbf{y}_p + r \frac{\mathbf{w}^{*T} \mathbf{w}^*}{\|\mathbf{w}^*\|} + w_0^* \\ g(\mathbf{y}) &= g(\mathbf{y}_p) + r \frac{\|\mathbf{w}^*\|^2}{\|\mathbf{w}^*\|} \\ r &= \frac{g(\mathbf{y})}{\|\mathbf{w}^*\|}\end{aligned}$$

O valor de r é positivo se \mathbf{y} está do lado positivo do hiperplano ótimo, e negativo se \mathbf{y} está do lado oposto. A distância do hiperplano para a origem é dada por $\frac{w_0^*}{\|\mathbf{w}^*\|}$. A figura 3.2 ilustra o hiperplano ótimo e as amostras em um espaço bidimensional.

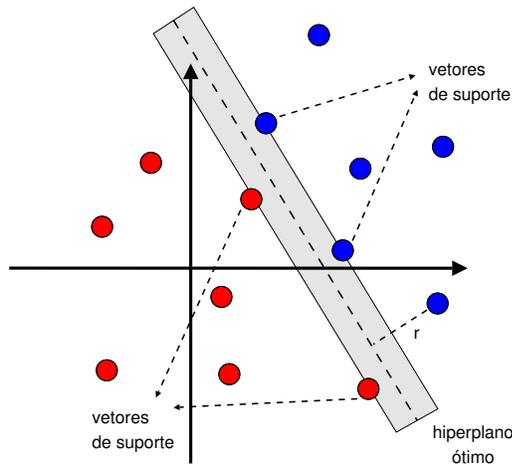


Figura 3.2: Ilustração do hiperplano ótimo em um espaço bidimensional.

Para encontrar o hiperplano ótimo, vamos assumir inicialmente que as amostras \mathbf{x}_i são linearmente separáveis, e d_i é a resposta esperada para uma amostra (e.g., $d_i = -1$ para uma amostra da classe ω_1 , $d_i = 1$ para uma amostra da classe ω_2). O par (\mathbf{w}^*, w_0^*) deve satisfazer a restrição

$$\begin{aligned}\mathbf{w}^{*T} \mathbf{y}_i + w_0^* &\geq 1 \quad \text{para } d_i = 1 \\ \mathbf{w}^{*T} \mathbf{y}_i + w_0^* &\leq -1 \quad \text{para } d_i = -1.\end{aligned}$$

Amostras que satisfazem a primeira ou a segunda linha com o sinal de igualdade são chamadas de vetores de suporte. Os vetores de suporte são pontos que estão mais próximos da superfície de decisão, e portanto mais difíceis de se classificar. O hiperplano ótimo é aquele que minimiza $\frac{\mathbf{w}^T \mathbf{w}}{2}$ satisfazendo a restrição $d_i(\mathbf{w}^T \mathbf{y}_i + w_0) \geq 1$ para $i = 1, \dots, N_{tr}$. Usando o método dos multiplicadores de Lagrange, temos

$$J(\mathbf{w}, w_0, \alpha) = \frac{\mathbf{w}^T \mathbf{w}}{2} - \sum_{i=1}^{N_{tr}} \alpha_i [d_i(\mathbf{w}^T \mathbf{y}_i + w_0) - 1]$$

onde as variáveis $\alpha_i \geq 0$ são os multiplicadores de Lagrange. Com as condições

$$\begin{aligned} \frac{\partial J(\mathbf{w}, w_0, \alpha)}{\partial \mathbf{w}} &= 0 \\ \frac{\partial J(\mathbf{w}, w_0, \alpha)}{\partial w_0} &= 0 \end{aligned}$$

chegamos à

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^{N_{tr}} \alpha_i d_i \mathbf{y}_i \\ \sum_{i=1}^{N_{tr}} \alpha_i d_i &= 0 \end{aligned}$$

Os valores ótimos de α_i são encontrados maximizando $Q(\alpha)$,

$$Q(\alpha) = \sum_{i=1}^{N_{tr}} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_{tr}} \sum_{j=1}^{N_{tr}} \alpha_i \alpha_j d_i d_j \mathbf{y}_i^T \mathbf{y}_j$$

sujeito às restrições

$$\sum_{i=1}^{N_{tr}} \alpha_i d_i = 0 \quad (3.9)$$

$$\alpha_i \geq 0 \quad (3.10)$$

Os vetores de suporte possuem $\alpha_i > 0$. Encontrados os multiplicadores de Lagrange ótimos, o vetor de pesos \mathbf{w}^* é dado por

$$\mathbf{w}^* = \sum_{i=1}^{N_s} \alpha_i d_i \mathbf{y}_i \quad (3.11)$$

onde N_s é o número de vetores de suporte, e o peso w_0^* pode ser encontrado com

$$w_0^* = 1 - \mathbf{w}^{*T} \mathbf{y}_s \quad (3.12)$$

para algum vetor de suporte \mathbf{y}_s .

Na prática, as amostras podem não ser linearmente separáveis. Este problema é contornado com uma modificação na restrição dada pela equação 3.10. Em vez de $\alpha_i \geq 0$, usamos $0 \leq \alpha_i \leq C$, onde C é uma constante positiva escolhida pelo usuário.

Como estamos interessados apenas em $\alpha_i > 0$, podemos reenumerar os multiplicadores e a equação 3.11 fica

$$\mathbf{w}^* = \sum_{j=1}^{N_s} \alpha_j d_j \varphi(\mathbf{x}_j) \quad (3.13)$$

onde $\varphi(\mathbf{x}_j) = [\varphi_1(\mathbf{x}_j), \dots, \varphi_m(\mathbf{x}_j)]^T$. O hiperplano ótimo pode ser reescrito como

$$\mathbf{w}^{*T} \varphi(\mathbf{x}) + w_0^* = 0 \quad (3.14)$$

onde $\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \dots, \varphi_m(\mathbf{x})]^T$. Substituindo a equação 3.13 na equação 3.14 temos

$$\sum_{j=1}^{N_s} \alpha_j d_j \varphi^T(\mathbf{x}_j) \varphi(\mathbf{x}) + w_0^* = 0 \quad (3.15)$$

onde $\varphi^T(\mathbf{x}_j) \varphi(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_j)$ representa o produto interno de dois vetores no espaço de características oculto, sendo \mathbf{x} o vetor de entrada e \mathbf{x}_j um vetor de suporte. A função $K(\mathbf{x}, \mathbf{x}_j)$, denominada núcleo do produto interno, é simétrica e deve satisfazer o teorema de Mercer [28]. Uma função comum e utilizada na pesquisa é a função de base radial, definida como

$$K(\mathbf{x}, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_j\|^2) \quad (3.16)$$

Dado $K(\mathbf{x}, \mathbf{x}_j)$, não é necessário o cálculo de φ e \mathbf{w}^* para a construção do hiperplano ótimo

$$\mathbf{w}^{*T} \varphi(\mathbf{x}) + w_0^* = \sum_{j=1}^{N_s} \alpha_j d_j K(\mathbf{x}, \mathbf{x}_j) + w_0^* = 0 \quad (3.17)$$

ou seja, basta verificar o sinal da equação $\sum_{j=1}^{N_s} \alpha_j d_j K(\mathbf{x}, \mathbf{x}_j) + w_0^*$ para saber se \mathbf{x} está do lado positivo ou negativo do hiperplano. A figura 3.3 ilustra a arquitetura de uma máquina de vetores de suporte, onde os pesos da rede são $\alpha_j d_j$. Quando temos múltiplas classes, cada suporte \mathbf{x}_j definirá um peso diferente para cada nó de saída, pois os pesos $\alpha_j d_j$ serão diferentes para cada classe, e a classe selecionada é aquela com maior $g_i(\mathbf{x})$.

3.3 Classificador Floresta de Caminhos Ótimos

Técnicas de aprendizado não-supervisionado normalmente interpretam as amostras do conjunto de treinamento como nós de um grafo completo (i.e., todos os pares de nós estão

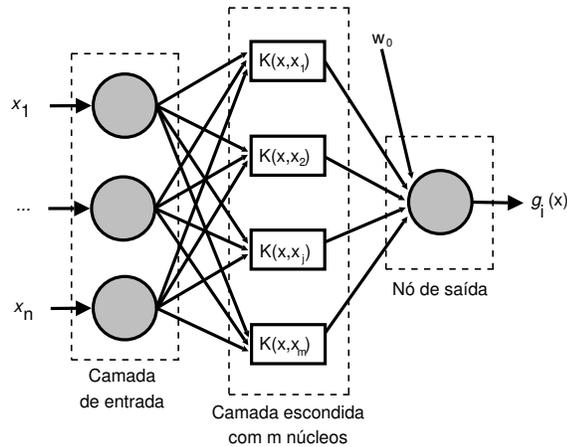


Figura 3.3: Arquitetura de uma máquina de vetores de suporte com m núcleos.

conectados por uma aresta). Neste caso, as amostras são representadas por vetores de características e o peso de uma aresta é a distância entre os vetores de características dos nós conectados pela aresta [46]. Estas técnicas visam particionar o grafo em componentes (*clusters*), de modo que cada *cluster* contenha somente amostras de uma mesma classe. No entanto, sem nenhum conhecimento prévio das classes não há garantia de que todos objetos em um dado *cluster* sejam da mesma classe.

O trabalho desenvolvido explora o mesmo conceito de interpretar o conjunto de treinamento como um grafo completo, usado para o aprendizado supervisionado e classificação. Nós definimos uma função de custo de caminho que associa a um caminho no grafo o custo de considerar todas as amostras ao longo do caminho como pertencentes a uma mesma classe. O algoritmo da IFT é usado para particionar o grafo em uma floresta de caminhos ótimos (OPF, do inglês *Optimum-Path Forest*) enraizada em nós protótipos, e cada protótipo define uma árvore de caminhos ótimos, cujos nós são mais “fortemente conexos” ao nó protótipo do que a qualquer outra raiz, de acordo com a função de custo de caminho definida.

O treinamento consiste na construção da floresta de caminhos ótimos, na qual os nós de uma árvore de caminhos ótimos são da mesma classe do nó protótipo. Para a classificação de uma amostra do conjunto de teste, a amostra é considerada parte do grafo de treinamento, e são avaliados todos os caminhos ótimos dos protótipos para a amostra. A classe do protótipo que oferece o melhor custo de caminho é atribuída à amostra de teste.

Neste trabalho nós utilizamos a função de custo f da equação 2.2, presente na seção sobre a IFT (Seção 2.3). Um conjunto de nós protótipos \mathcal{S}^* é definido como ótimo quando o classificador acerta a classificação de todas as amostras do conjunto de treinamento \mathbf{Z}_{tr} . Nós podemos encontrar \mathcal{S}^* através da relação teórica entre a árvore geradora mínima

(MST, do inglês *Minimum Spanning Tree*) [8] e a árvore de caminhos ótimos para a função de custo da equação 2.2. Ao calcular a MST no grafo completo, nós obtemos um grafo conexo acíclico não orientado, cujos nós são todas as amostras em \mathbf{Z}_{tr} e o peso de uma aresta é a distância entre nós adjacentes. Esta árvore geradora é ótima porque a soma dos pesos de todas as arestas é mínima quando comparada com qualquer outra árvore geradora no grafo completo. Na MST, todo par de nós está conectado por um único caminho ótimo. Para qualquer nó do grafo de treinamento, é possível direcionar as arestas da MST de modo a obter uma árvore de caminhos ótimos enraizada nesse nó. Os nós protótipos ótimos são todos os nós da MST conectados por uma aresta e que pertencem a classes diferentes. Nesta floresta de caminhos ótimos, cada protótipo funciona como uma barreira, evitando que nós pertencentes a sua árvore sejam conquistados por outros protótipos.

A distância entre nós adjacentes foi definida como a distância Euclideana entre os vetores de características dos nós correspondentes. A figura 3.4 ilustra os passos do algoritmo.

3.3.1 Algoritmo de aprendizado

O classificador tem melhor desempenho quando amostras mais próximas, de classes diferentes, são incluídas no conjunto de treinamento. No melhor caso, se todas as amostras mais próximas, de classes diferentes, forem incluídas no conjunto de treinamento, o classificador acertará a classificação de todas as amostras de teste. Como não é possível garantir a presença de todas essas amostras no conjunto de treinamento, podemos melhorar o desempenho do classificador através do retreinamento com as amostras classificadas incorretamente do conjunto de teste.

Para avaliar o desempenho de um classificador, definimos uma medida que leva em consideração o fato de que as classes podem ter quantidades diferentes de amostras. Sendo N_{tr} o número de amostras no conjunto de treinamento \mathbf{Z}_{tr} , e N_{te} o número de amostras no conjunto de teste $\mathbf{Z}_{te} = W_1 \cup W_2 \cup \dots \cup W_c$, onde W_i é o conjunto de amostras de teste da classe i e c o número de classes, podemos definir

$$e_{i,1} = \frac{FP_i}{N_{te} - |W_i|} \quad \text{e} \quad e_{i,2} = \frac{FN_i}{|W_i|}, \quad i = 1, \dots, c \quad (3.18)$$

onde FP_i e FN_i são as taxas de falsos positivos e falsos negativos, respectivamente. FP_i é o número de amostras que foram classificadas incorretamente como pertencentes à classe i , e FN_i é o número de amostras da classe i que foram incorretamente classificadas como pertencentes à outra classe. Os erros $e_{i,1}$ e $e_{i,2}$ são usados para definir

$$E_i = e_{i,1} + e_{i,2} \quad (3.19)$$

onde E_i é a soma parcial dos erros da classe i . A acurácia Acc do classificador pode ser definida como

$$Acc = \frac{2c - \sum_{i=1}^c E_i}{2c} = 1 - \frac{\sum_{i=1}^c E_i}{2c} \quad (3.20)$$

Para avaliar a capacidade de um classificador aprender com os erros, tendo um conjunto de treinamento de tamanho limitado, nós desenvolvemos um algoritmo de aprendizado que retreina o classificador com as amostras classificadas incorretamente na iteração anterior e constrói uma curva com as medidas de acurácia de cada iteração.

Algoritmo 2. ALGORITMO DE APRENDIZADO

ENTRADA: Conjunto de treinamento \mathbf{Z}_{tr} , conjunto de teste \mathbf{Z}_{te} e número de iterações T .
 SAÍDA: Curva de aprendizado L .
 AUXILIARES: Vetores FP e FN de tamanho c

1. **Para** $i = 1$ até c **faça**
2. $FN[i] \leftarrow 0$ e $FP[i] \leftarrow 0$.
3. **Para** $i = 1$ até T **faça**
4. Encontre o conjunto de nós protótipos \mathcal{S}^* .
5. Construa o classificador com o conjunto de treinamento \mathbf{Z}_{tr} .
6. **Para cada** $z \in \mathbf{Z}_{te}$ **faça**
7. Seja j a classe atribuída a z pelo classificador.
8. **Se** $j \neq l(z)$, **então**
9. $FN[l(z)] \leftarrow FN[l(z)] + 1$.
10. $FP[j] \leftarrow FP[j] + 1$.
11. Marque z como uma amostra classificada incorretamente.
12. $L[i] \leftarrow Acc$ (Equação 3.20).
13. Substitua aleatoriamente amostras em \mathbf{Z}_{tr} que não são protótipos por amostras classificadas incorretamente de mesma classe em \mathbf{Z}_{te} .

Os nós protótipos preservados na linha 13 podem não ser escolhidos como protótipos (linha 4) na próxima iteração, e assim podem ser substituídos por amostras classificadas incorretamente na iteração seguinte.

Ao limitar o tamanho de \mathbf{Z}_{tr} em uma determinada porcentagem de $N_{tr} + N_{te}$, podemos também manter o número de amostras de cada classe em \mathbf{Z}_{tr} como uma porcentagem do total de amostras da classe. O tamanho fixo de \mathbf{Z}_{tr} impõe um limite na capacidade de aprendizado do classificador. Nós avaliamos a capacidade de aprendizado LC com a equação

$$LC = 1 - \frac{|\mathcal{S}^*|}{N_{tr}} \quad (3.21)$$

Quanto maior o número de protótipos em \mathbf{Z}_{tr} , menor é a capacidade de aprendizado. Se $LC = 0$, então a linha 13 do algoritmo não substituirá nenhuma amostra, a curva

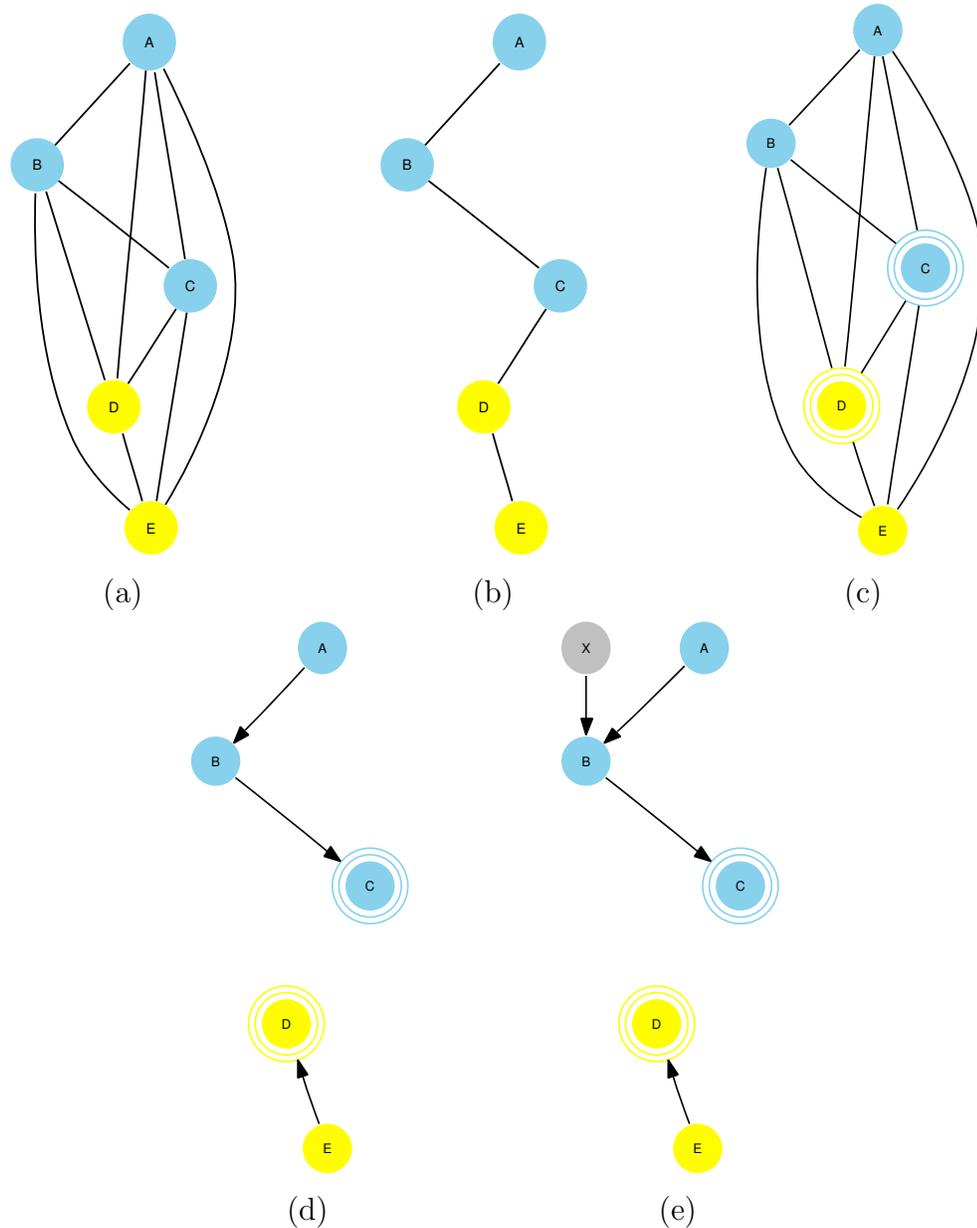


Figura 3.4: Exemplo de treinamento e teste do classificador Floresta de Caminhos Ótimos. (a) Grafo completo de treinamento com cinco amostras. Os nós A, B e C são da classe ω_1 , e os nós D e E são da classe ω_2 . (b) Árvore geradora mínima executada sobre o grafo completo. (c) Os nós C e D são de classes diferentes e estão ligados por uma aresta na árvore geradora mínima. Portanto, estes nós se tornam protótipos no grafo completo. (d) A partir dos protótipos é calculada uma floresta de caminhos ótimos. Os nós C e D são raízes das árvores de caminhos ótimos. (e) Para a classificação de uma amostra de teste X, o classificador avalia os caminhos dos protótipos até a amostra. A classe atribuída à amostra é a classe do protótipo que oferece o caminho ótimo para esta amostra.

de aprendizado ficará constante e o melhor desempenho foi alcançado para o tamanho do conjunto de treinamento. Um valor baixo de LC pode indicar que os vetores de características são ruins, e não conseguem diferenciar de forma eficiente as amostras de classes diferentes.

A curva de aprendizado também descreve a velocidade com que o classificador aprende com os erros de classificação das amostras de teste. Esta curva pode ser usada para a comparação de classificadores, porque mostra a variação de acurácia dos classificadores para vários conjuntos de treinamento.

O algoritmo de aprendizado pode ser usado com os classificadores Rede Neural e SVM. Apesar de não possuírem o conceito de nós protótipos, o algoritmo pode ser modificado de modo a manter o desempenho desses classificadores ao longo das iterações. Na rede neural, os pesos \mathbf{w} dos nós da rede na iteração atual são usados para a inicialização da rede na iteração seguinte. No SVM, as amostras relacionadas aos vetores de suporte são mantidas em \mathbf{Z}_{tr} .

As figuras 3.5a e 3.5b mostram as curvas de aprendizado dos classificadores Rede Neural, SVM e OPF, para a base de dados MPEG-7 [1], com 70 classes, e uma base sintética “meio-anel” [30], com 2 classes, ilustrada na figura 3.5c. A base sintética possui 46.463 amostras e uma classe dez vezes maior que a outra. Para a base MPEG-7 foi usado o descritor Dimensão Fractal Multiescala [12]. A rede neural para a base MPEG-7 possui 10 nós de entrada, 5 nós na camada escondida e 70 nós de saída, e para a base “meio-anel” possui 2 nós de entrada, 5 nós na camada escondida e 2 nós de saída. Os conjuntos de treinamento e teste foram limitados em 50% do número de amostras.

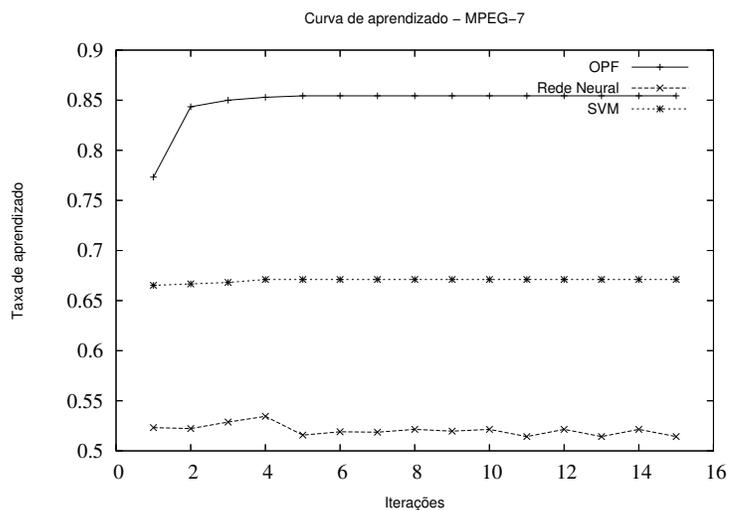
A oscilação da curva para a rede neural indica a instabilidade do classificador, mesmo com os pesos sinápticos mantidos entre as iterações. A curva do classificador SVM se estabiliza em um valor de acurácia inferior ao valor do classificador OPF. Para a base “meio-anel”, após 15 iterações foram estabelecidos 2 protótipos para o classificador OPF e 4.619 vetores de suporte para o classificador SVM.

3.4 Coleção de classificadores

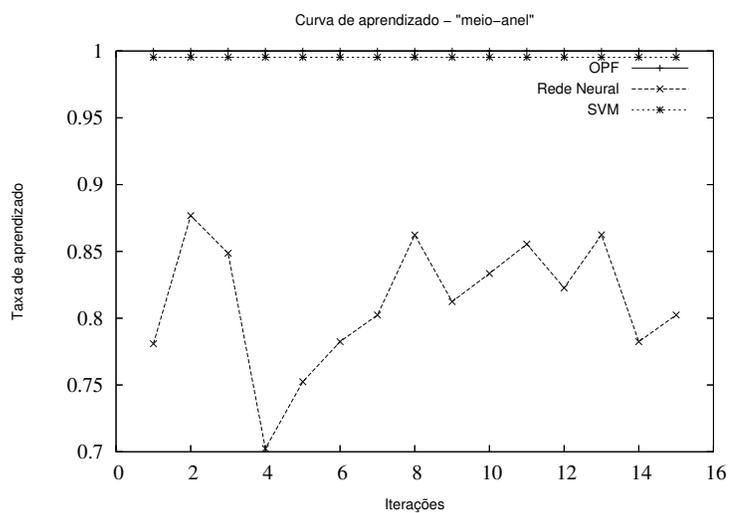
As técnicas de aprendizado para coleção de classificadores são usadas para melhorar a estabilidade e desempenho dos classificadores. As técnicas *Bagging* e *Adaboost* são técnicas populares para coleção de classificadores, e são descritas a seguir:

3.4.1 *Bagging*

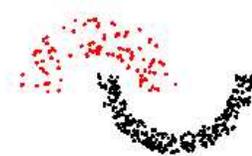
A técnica *Bagging* (*Bootstrap aggregating*) [4, 17] usa múltiplas versões do conjunto de treinamento, onde cada versão é criada através da amostragem aleatória com reposição



(a)



(b)



(c)

Figura 3.5: (a) Curva de aprendizado para a base MPEG-7. (b) Curva de aprendizado para a base “meio-anel”. (c) Ilustração da base sintética “meio-anel”.

de elementos em \mathbf{Z}_{tr} . Devido à reposição, as amostras de \mathbf{Z}_{tr} podem se repetir em cada versão criada. Cada classificador é treinado com uma versão do conjunto de treinamento, e contribui com um voto para a decisão final. A classe escolhida é aquela com o maior número de votos.

Um ponto importante para a melhoria de desempenho é a instabilidade dos classificadores combinados. Se um conjunto \mathbf{Z}'_{tr} modificado (e.g., com amostras repetidas), produzir pequenas mudanças no resultado do classificador treinado, então este classificador será próximo do classificador treinado com \mathbf{Z}_{tr} , e a técnica *Bagging* será ineficiente. A melhoria de desempenho ocorre com classificadores instáveis (e.g., redes neurais, árvores de decisão), onde pequenas mudanças no conjunto de treinamento produzem grandes mudanças no resultado do classificador treinado.

3.4.2 *Adaboost*

A idéia da técnica *Adaboost* [25,30] é formar incrementalmente o grupo de classificadores, adicionando um classificador por vez. Cada amostra do conjunto de treinamento recebe um peso que determina a probabilidade dessa amostra ser selecionada para o treinamento de um classificador. Se a amostra for classificada corretamente, a chance de ser usada novamente para o treinamento de outro classificador é reduzida. Caso contrário, a chance de ser usada novamente é aumentada. Desta forma, a técnica se concentra em amostras de difícil classificação. Inicialmente, os pesos das amostras são uniformes. A cada iteração k , um conjunto de treinamento é formado com amostras selecionadas de \mathbf{Z}_{tr} , de acordo com os pesos, e é usado para o treinamento de um classificador C_k . As amostras de \mathbf{Z}_{tr} classificadas incorretamente por C_k têm o peso incrementado, e amostras classificadas corretamente têm o peso reduzido. Um peso é atribuído ao classificador, e o processo é repetido até se alcançar um limite de erro ou um limite no número de classificadores treinados. O algoritmo abaixo é usado para o treinamento dos classificadores:

Algoritmo 3. ADABOOST - TREINAMENTO

ENTRADA: Conjunto de treinamento \mathbf{Z}_{tr} , número de amostras N_{tr} e número de iterações K .
 SAÍDA: Coleção de classificadores \mathbf{C} e seus pesos β .
 AUXILIARES: Vetor de erros \mathbf{E} , vetor de pesos \mathbf{w}

1. **Para** $i = 1$ até N_{tr} **faça**
2. $w^1[i] \leftarrow \frac{1}{N_{tr}}$
3. $\mathbf{C} \leftarrow \emptyset$
4. **Para** $k = 1$ até K **faça**
5. | Amostre um conjunto de treinamento \mathbf{Z}_{tr}^k de \mathbf{Z}_{tr} usando distribuição \mathbf{w}^k .
6. | Treine um classificador C_k com o conjunto de treinamento \mathbf{Z}_{tr}^k .

7. $\left| \begin{array}{l} \text{Calcule o erro do classificador na iteração } k \text{ por } E[k] \leftarrow \sum_{j=1}^{N_{tr}} w^k[j] l_j^k, \\ \text{onde } l_j^k \leftarrow 1 \text{ se } C_k \text{ classificar incorretamente } z_j, \text{ e } l_j^k \leftarrow 0 \text{ no caso contrário.} \end{array} \right.$
8. **Se** $E[k] = 0$ **ou** $E[k] \geq 0.5$ **então**
9. $\left| \begin{array}{l} \text{Descarte } C_k. \end{array} \right.$
10. $\left[\begin{array}{l} \text{Para } i = 1 \text{ até } N_{tr} \text{ faça} \\ \text{L } w^k[i] \leftarrow \frac{1}{N_{tr}} \end{array} \right.$
11. $\left[\begin{array}{l} \text{L } w^k[i] \leftarrow \frac{1}{N_{tr}} \end{array} \right.$
12. **senão**
13. $\left| \begin{array}{l} \mathbf{C} \leftarrow \mathbf{C} \cup C_k \\ \beta[k] \leftarrow \frac{E[k]}{1-E[k]} \end{array} \right.$
14. $\left[\begin{array}{l} \beta[k] \leftarrow \frac{E[k]}{1-E[k]} \end{array} \right.$
15. $\left[\begin{array}{l} \text{Para } j = 1 \text{ até } N_{tr} \text{ faça} \\ \text{L } w^{k+1}[j] = \frac{w^k[j] \beta[k]^{(1-l_j^k)}}{\sum_{i=1}^{N_{tr}} w^k[i] \beta[k]^{(1-l_i^k)}} \end{array} \right.$
16. $\left[\begin{array}{l} \text{L } \left[\begin{array}{l} \text{L } w^{k+1}[j] = \frac{w^k[j] \beta[k]^{(1-l_j^k)}}{\sum_{i=1}^{N_{tr}} w^k[i] \beta[k]^{(1-l_i^k)}} \end{array} \right. \end{array} \right.$

Para a classificação de uma amostra é necessário o cálculo dos suportes das classes. O suporte para uma classe ω_t é dado por

$$\mu_t(z) = \sum_{C_k(z)=\omega_t} \log \frac{1}{\beta_k} \quad (3.22)$$

A classe com o maior suporte é atribuída à amostra.

Capítulo 4

Base de Imagens

Uma base de imagens de parasitos está sendo criada pelo aluno de doutorado em Biologia, para o desenvolvimento e validação dos métodos pesquisados.

A seção 4.1 comenta o sistema de aquisição das imagens. A seção 4.2 descreve as características dos parasitos trabalhados, e a seção 4.3 detalha a base de imagens em formação.

4.1 Aquisição das imagens

O sistema utilizado para a aquisição das imagens é composto pelos seguintes elementos:

1. Kit para exame parasitológico das fezes;
2. Bomba peristáltica;
3. Microscópio;
4. Câmera digital;
5. Computador.

As amostras fecais são coletadas e armazenadas em um kit específico para exame parasitológico. Estas amostras são processadas em laboratório, até a obtenção de sedimento fecal em um tubo de centrifugação. Os tubos de centrifugação contendo sedimento processado e homogeneizado são colocados nos suportes da bomba peristáltica, que realiza o resgate de uma pequena quantidade de sedimento fecal e envia para a lâmina localizada no microscópio. A câmera digital acoplada ao microscópio captura as imagens, que são armazenadas em um computador. Este processo é repetido até que todo o sedimento fecal no tubo de centrifugação seja resgatado. Atualmente, a regulagem e acionamento

dos dispositivos estão sendo realizados manualmente, e a automatização da aquisição das imagens será realizada posteriormente com o uso de platina motorizada e *driver* de foco. A figura 4.1 ilustra o sistema com aquisição automática de imagens, onde o computador controla o acionamento da bomba peristáltica e os outros dispositivos para a captura das imagens:

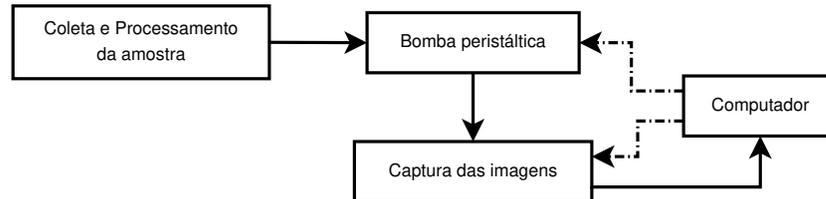


Figura 4.1: Sistema com aquisição automática de imagens. As setas normais indicam a seqüência das etapas para a aquisição das imagens, e as setas tracejadas indicam sinais de controle dos equipamentos. O computador controla a bomba peristáltica e os dispositivos para a captura das imagens.

4.2 Parasitos

Os parasitos intestinais do Homem podem ser separados em duas categorias: protozoários e helmintos. Os helmintos podem ser subdivididos em ovos e larvas. Os parasitos existentes em cada categoria são:

1. Protozoários

(a) Patogênicos

Entamoeba histolytica, *Giardia intestinalis*, *Blastocystis hominis* (patogenicidade controversa), *Cryptosporidium spp.*, *Cyclospora cayetanensis*, e *Isospora belli*.

(b) Não patogênicos

Entamoeba dispar, *Entamoeba coli*, *Entamoeba hartmanni*, *Endolimax nana*, *Iodamoeba bütschlii*, *Chilomastix mesnili*, *Dientamoeba fragilis*, *Trichomonas hominis*, e *Balantidium coli*.

2. Helmintos

(a) Ovos

Ascaris lumbricoides, *Enterobius vermiculares*, ancilostomatídeos, *Trichuris trichiura*, *Hymenolepis nana*, *Hymenolepis diminuta*, *Taenia spp.* (*Taenia saginata*/*Taenia solium*), *Schistosoma mansoni*, *Schistosoma haematobium*, *Schistosoma japonicum*, *Diphyllobothrium latum*, *Clonorchis sinensis*, *Paragonimus westermani*, *Fasciolopsis buski*, *Fasciola hepatica*, *Metagonimus yokogawai*, e *Heterophyes heterophyes*.

(b) Larvas

Ancilostomatídeos e *Strongyloides stercoralis*.

Alguns desses parasitos não são comuns no Brasil e outros são muito raros. Por isso, a meta inicial foi concentrar o trabalho nos parasitos de maior prevalência no país. As tabelas 4.1 e 4.2 listam esses parasitos, as doenças que causam, forma e tamanho de cada espécie [14].

Espécie	Doença	Forma	Comprim./Larg. (μm)
<i>Entamoeba histolytica</i> / <i>E. dispar</i>	amebíase	Esférico	10-20 (média 12-15)
<i>Giardia intestinalis</i>	giardíase	Elipsóide, esférico ou oval	8-19 / 7-10
<i>Isospora belli</i>	isosporíase	Elipsóide	20-30 / 10-19
<i>Entamoeba coli</i>	não patogênico	Esférico	10-35 (média 15-25)
<i>Endolimax nana</i>	não patogênico	Esférico	5-10 (média 6-8)
<i>Iodamoeba bütschlii</i>	não patogênico	Esférico	5-20 (média 10-12)
<i>Blastocystis hominis</i>	patogenicidade controversa	Elipsóide, esférico ou oval	4-15 (média 8-10)

Tabela 4.1: Protozoários intestinais mais comuns, doenças que causam, formas e tamanhos.

Nota-se a diferença de tamanho entre protozoários e helmintos, onde o comprimento dos ovos de helmintos pode chegar a mais de dez vezes o tamanho dos protozoários, e as larvas possuem forma totalmente diferente dos outros parasitos.

Uma das razões para erro no diagnóstico dos métodos tradicionais é que o microscopista analisa a lâmina com sedimento fecal com objetiva de aumento de 100 vezes, e só passa para aumento de 400 vezes quando há suspeita de presença de parasitos. Com isso, muitos protozoários podem ser confundidos com impurezas e passar despercebidos na análise com aumento de 100 vezes. Além disso, somente uma pequena quantidade do sedimento fecal é separado para análise (aproximadamente 1/12), o que leva também a aumentar a possibilidade de erro. Considerando estes aspectos, todas as imagens utilizadas no projeto

Espécie	Doença	Forma	Comprim./Larg. (μm)
<i>Ascaris lumbricoides</i>	ascaridíase	Oval	55-75 / 35-50 (fértil) 85-95 / 43-47 (infértil)
<i>Enterobius vermicularis</i>	enterobíase ou oxiuríase	Oval	50-60 / 20-30
Ancilostomatídeos	ancilostomíase	Oval	60-75 / 36-40
<i>Strongyloides stercoralis</i>	estrongiloidíase	Rabditóide	180-380 / 14-20
<i>Trichuris trichiura</i>	tricuríase	Aspecto típico de barril	50-55 / 20-23
<i>Hymenolepis nana</i>	himenolopíase	Esférico	30-47
<i>Hymenolepis diminuta</i>	himenolopíase	Oval	70-85 / 60-80
<i>Taenia saginata</i>	teníase	Esférico	31-43
<i>Taenia solium</i>	teníase e cisticercose	Esférico	31-43
<i>Schistosoma mansoni</i>	esquistossomíase	Oval	114-175 / 45-70

Tabela 4.2: Helmintos intestinais mais comuns, doenças que causam, formas e tamanhos.

foram obtidas com aumento de 400 vezes, e o sistema de aquisição automática de imagens permitirá a leitura de todo o sedimento fecal. Desta forma, o sistema permitirá a análise de uma quantidade de imagens superior à capacidade humana, com taxa de erro menor que as obtidas com técnicas tradicionais.

As figuras 4.2 e 4.3 ilustram as espécies de protozoários e helmintos mais comuns.

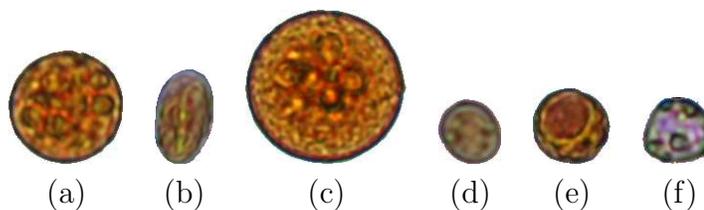


Figura 4.2: Exemplo de espécies de protozoários mais comuns. (a) *Entamoeba histolytica*; (b) *Giardia intestinalis*; (c) *Entamoeba coli*; (d) *Endolimax nana*; (e) *Iodamoeba bütschlii*; (f) *Blastocystis hominis*.

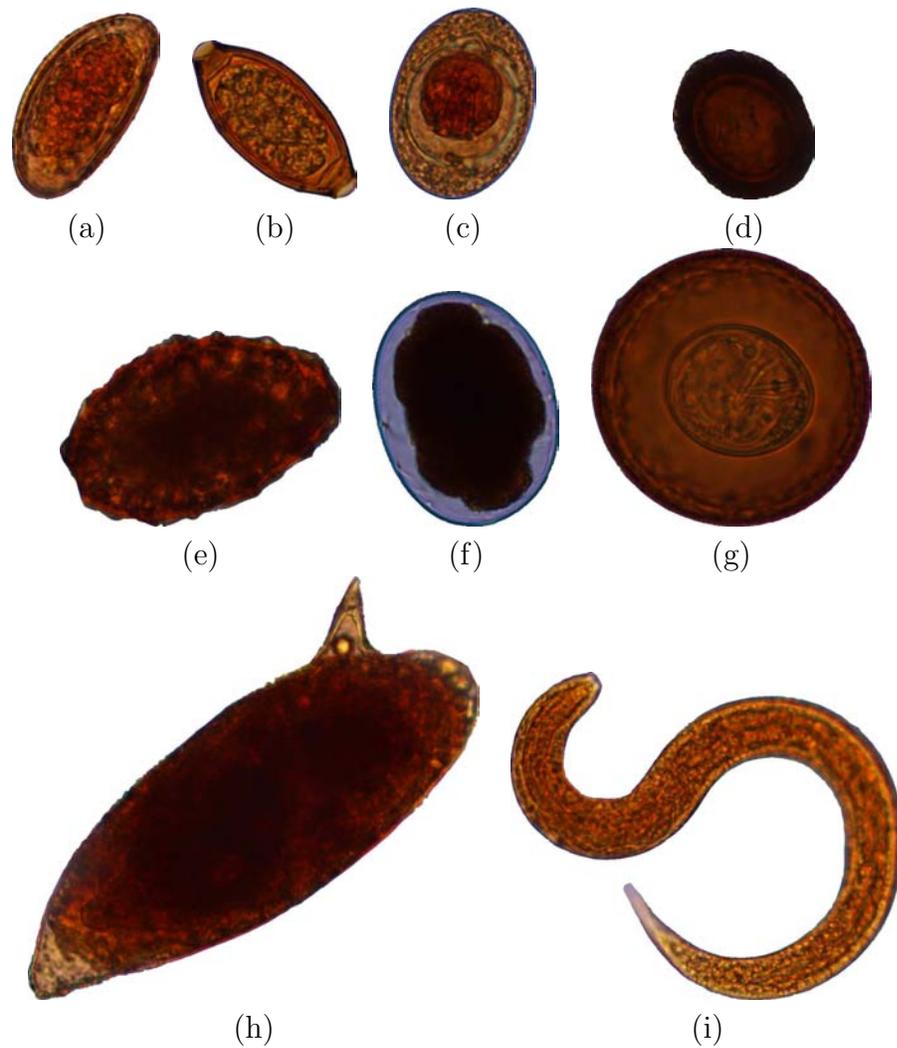


Figura 4.3: Exemplo de espécies de helmintos mais comuns. (a) *Enterobius vermiculares*; (b) *Trichuris trichiura*; (c) *Hymenolepis nana*; (d) *Taenia spp.*; (e) *Ascaris lumbricoides*; (f) Ancilostomatídeos; (g) *Hymenolepis diminuta*; (h) *Schistosoma mansoni*; (i) Larva de *Strongyloides stercoralis*.

4.3 Bases de dados para projeto e teste

As amostras fecais são coletadas em diversas regiões do Estado de São Paulo: Hospital de Clínicas da Universidade Estadual de Campinas (Unicamp); Hospital de Clínicas da Universidade Estadual Paulista (Unesp), Botucatu; Delegacia Regional da Saúde do Município de Botucatu (DIR XI); e Hospital Pronto Socorro Ouro Verde da cidade de Campinas. Todos os processamentos e triagens dessas amostras são realizados no Laboratório de Parasitologia do Hospital de Clínicas da Unicamp. Uma solução corante foi desenvolvida especificamente para o projeto, de forma a criar um bom contraste entre o fundo da imagem, de cor azul, e os parasitos, que adquirem um tom castanho. As imagens foram capturadas com aumento de 400 vezes, totalizando 455 parasitos em 407 imagens.

As imagens de parasitos intestinais presentes na base são de 16 espécies: 68 ovos de *Ascaris lumbricoides*; 12 ovos de *Enterobius vermicularis*; 6 ovos de Ancilostomatídeos; 30 ovos de *Trichuris trichiura*; 12 ovos de *Hymenolepis nana*; 9 ovos de *Hymenolepis diminuta*; 46 ovos de *Taenia spp.*; 17 ovos de *Schistosoma mansoni*; 4 ovos de *Diphyllobothrium spp.*; 11 larvas de *Strongyloides stercoralis*; 8 cistos de *Entamoeba histolytica*; 128 cistos de *Giardia intestinalis*; 55 cistos de *Entamoeba coli*; 40 cistos de *Endolimax nana*; 6 cistos de *Iodameba bütschlii*; e 3 cistos de *Blastocystis hominis*. As imagens estão no formato JPEG, possuem resolução de 2560x1920 pixels (5 Megapixels), e cada uma cobre uma região da lâmina de 300 μ m de comprimento por 225 μ m de largura.

O método de segmentação de parasitos é dividido em delineamento e reconhecimento. O delineamento extrai componentes das imagens de entrada, que são usados no reconhecimento para a extração de características e classificação em parasitos ou impurezas.

Além da base de imagens de parasitos, há também uma base de componentes para projeto dos classificadores. Os 455 parasitos foram segmentados manualmente, e são usados para a avaliação da qualidade do delineamento automático e projeto dos classificadores. A base de projeto também possui 2.528 não-parasitos (impurezas e parasitos com falhas graves no delineamento) e 449 parasitos com pequenas falhas no delineamento, que foram obtidos ao longo dos experimentos com *pipelines* diferentes do método atual. As amostras desta base podem ser separadas em três grupos para o treinamento dos classificadores: protozoários, com 476 parasitos e 1579 não-parasitos de tamanho pequeno; ovos de helmintos, com 406 parasitos e 737 não-parasitos de tamanho grande; e larvas, com 22 parasitos e 212 não-parasitos com forma alongada. Esta base é dividida em 50% para treinamento dos classificadores e 50% para avaliação do treinamento.

O delineamento automático das 407 imagens gera 2.774 componentes, contendo parasitos e impurezas. Estes componentes são a base de teste para a avaliação do delineamento e reconhecimento.

Capítulo 5

Segmentação de imagens

Dada uma imagem de amostra fecal, podendo conter um ou mais parasitos, a segmentação consiste em particionar a imagem em regiões de pixels que representem parasitos e fundo da imagem. As dificuldades do trabalho são ilustradas na Seção 5.1, e o método desenvolvido é apresentado na Seção 5.2.

5.1 Apresentação do problema

O fundo da imagem de um exame parasitológico consiste de regiões planas e regiões com impurezas que aparecem na imagem com tamanhos da mesma ordem de grandeza, ou maiores que os tamanhos dos parasitos. As impurezas podem apresentar coloração, forma ou textura semelhantes aos parasitos, e podem estar junto ao contorno ou sobre os parasitos, cobrindo trechos de suas bordas. É possível observar, nas figuras 4.2 e 4.3, que há variação de forma e tamanho entre as espécies de parasitos, principalmente para os helmintos. As larvas apresentam um problema adicional, pois possuem forma alongada e partes do parasito podem se sobrepor, criando uma região de fundo de imagem dentro do parasito, como pode ser observado nas figuras 5.1a e 5.1b. A figura 5.1c mostra um exemplo de imagem com impurezas de vários tamanhos, e a figura 5.1d ilustra uma impureza com forma e coloração semelhantes ao parasito *Giardia intestinalis*. As figuras 5.1e e 5.1f mostram impurezas cobrindo trechos do contorno dos parasitos.

5.2 Descrição do Método

Um parasito na imagem é um objeto representado por um único componente conexo. A forma é o atributo global mais relevante para distinguir um parasito das impurezas presentes na imagem. Cor e textura são relevantes para eliminar o fundo da imagem, mas

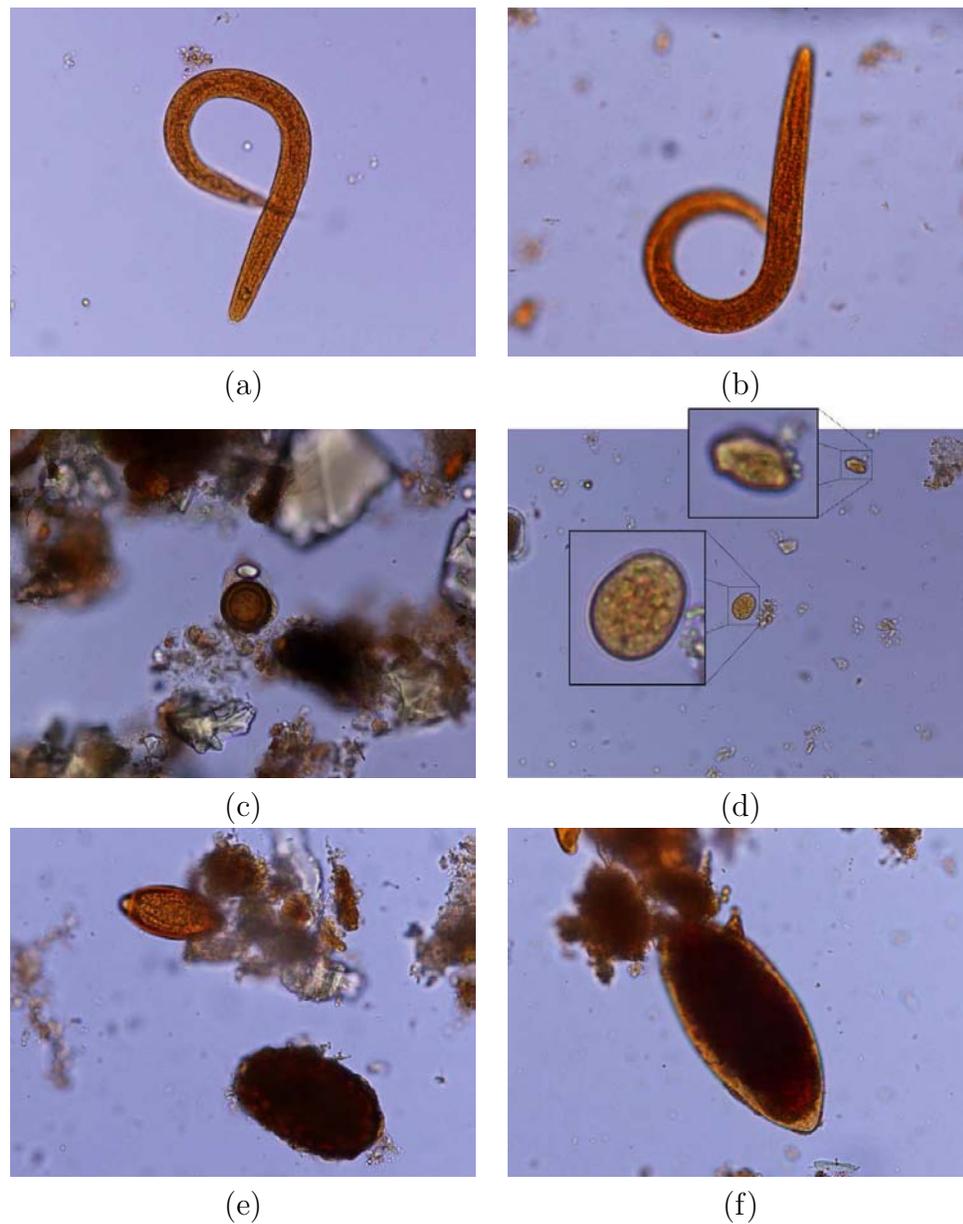


Figura 5.1: (a,b) Exemplos de larvas com partes do parasito que se sobrepõem, criando uma região de fundo de imagem no interior. (c) Exemplo de imagem de um ovo de *Taenia spp.* com impurezas de tamanhos variados; (d) Exemplo de imagem de uma *Entamoeba coli* com impureza de forma e coloração semelhante a uma *Giardia intestinalis*; (e) Exemplo de imagem de ovo de *Trichuris trichiura* com parte do contorno coberto por impurezas. A imagem apresenta também um ovo de *Ascaris lumbricoides* na parte inferior; (f) Exemplo de imagem de ovo de *Schistosoma mansoni* com parte do contorno coberto por impurezas.

muitas impurezas adquirem cor e textura similares a dos parasitos.

A segmentação de imagens consiste de duas tarefas fortemente acopladas: delimitação e reconhecimento [24]. O delimitação visa definir de forma precisa a extensão espacial do objeto na imagem. O reconhecimento está associado a subtarefas cognitivas, tais como a localização do objeto e a verificação do delimitação. O ser humano supera a máquina no reconhecimento, mas o contrário é verificado no delimitação. Técnicas eficientes de segmentação interativa normalmente exploram este fato [3, 18, 24, 37]. Na segmentação automática, o usuário pode ser substituído por técnicas que buscam transferir para a máquina a execução de tarefas cognitivas [6, 7, 44]. Uma questão importante é como combinar estas técnicas com outras de segmentação que são superiores na tarefa de delimitação.

O problema de separar os parasitos do resto da imagem foi dividido em dois subproblemas: o delimitação de componentes relevantes da imagem e o reconhecimento desses componentes como parasitos ou não-parasitos. Cada subproblema possui uma ou mais etapas de processamento, que são combinadas em seqüência formando um *pipeline*. A solução apresentada utiliza operadores baseados na IFT para o delimitação de componentes, e explora a característica de forma elíptica dos parasitos para localizar sementes da IFT. Os componentes delimitados (parasitos e impurezas) são usados para o reconhecimento. Esta solução alcançou desempenho muito superior a outras técnicas testadas, e por isso será apresentada somente a técnica com o melhor desempenho.

Analisando as características dos parasitos, podemos separá-los em três grupos: protozoários, com forma elíptica ou esférica e tamanho pequeno; ovos de helmintos, com forma esférica ou elíptica mais irregular do que a dos protozoários, e tamanho grande; e larvas, com forma alongada e tamanho grande. A estratégia com melhor desempenho foi a de criar um *pipeline* especialista para cada grupo, e processar cada imagem nos três *pipelines*. A figura 5.2 mostra a visão geral do *pipeline* para protozoários e ovos de helmintos, e a figura 5.3 mostra a visão geral do *pipeline* para larvas. Os *pipelines* para protozoários e ovos de helmintos possuem as mesmas etapas, mas com ajustes para componentes de tamanhos diferentes, e o *pipeline* para larvas trata casos em que partes do parasito se sobrepõem, criando uma região de fundo da imagem no interior do componente.

O delimitação de componentes e o reconhecimento são detalhados a seguir.

5.2.1 Delimitação de componentes

O delimitação visa extrair componentes conexos relevantes (parasitos e o mínimo possível de não-parasitos) das imagens. O método apresentado utiliza operadores baseados na IFT e identifica componentes com forma elíptica ou esférica para a estimativa de sementes. Apesar das larvas não terem a forma elíptica ou esférica, o método encontra elipses em

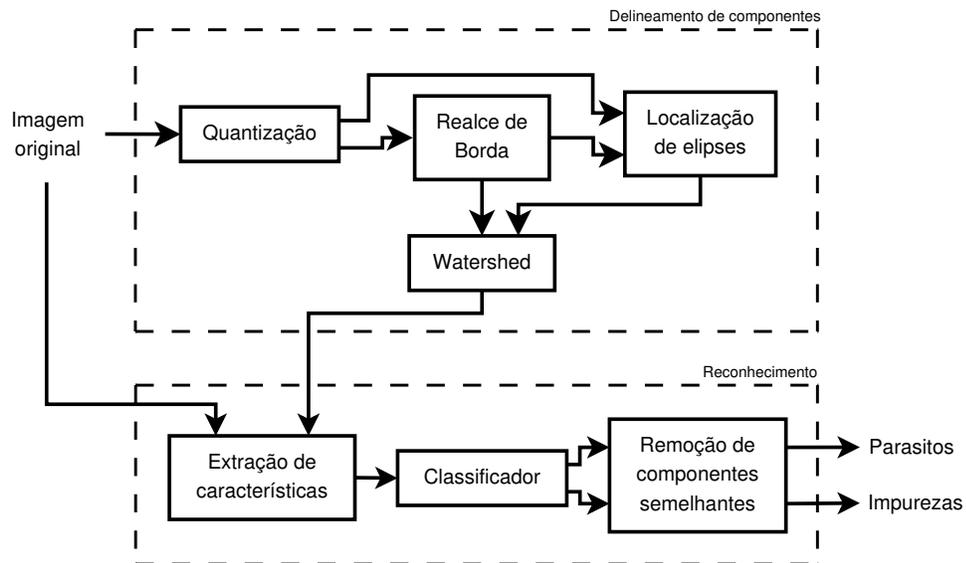


Figura 5.2: Etapas do *pipeline* para protozoários e ovos de helmintos.

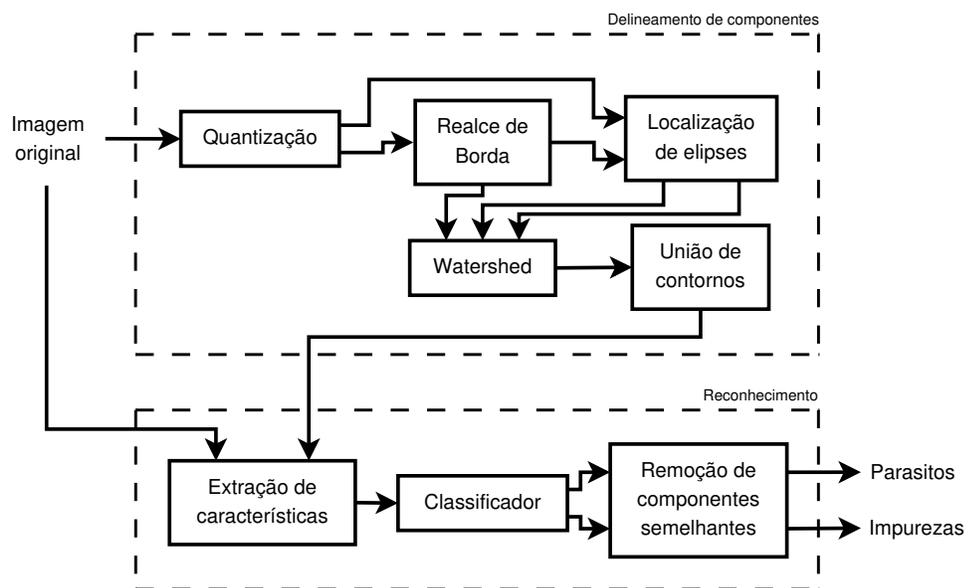


Figura 5.3: Etapas do *pipeline* para larvas de helmintos.

partes arredondadas do parasito, sendo suficientes para o delineamento da larva inteira.

O módulo de delineamento de componentes possui as seguintes etapas:

1. **Quantização** - Esta etapa converte a imagem original colorida para 64 níveis de cinza. O valor quantizado $I_q(p)$ de um dado pixel p é obtido pela equação

$$I_q(p) = \left\lfloor \frac{I_G(p)}{64} \right\rfloor + 4 \left\lfloor \frac{I_R(p)}{64} \right\rfloor + 16 \left\lfloor \frac{I_B(p)}{64} \right\rfloor \quad (5.1)$$

onde $I_R(p)$, $I_G(p)$ e $I_B(p)$ são os valores das bandas vermelha, verde e azul do pixel p . A etapa é usada para suavizar detalhes dentro e fora dos parasitos, sem eliminar o contorno dos componentes relevantes. A figura 5.5a ilustra a imagem 5.1e quantizada.

2. **Realce de borda** - Calcula a imagem de gradiente, com as bordas dos componentes realçadas, de acordo com a equação 2.4. A figura 5.5b ilustra a imagem 5.5a com as bordas realçadas.
3. **Localização de elipses** - Esta etapa é utilizada para localizar elipses em regiões da imagem com grande chance de serem componentes elípticos ou esféricos, e é ilustrada em detalhes na figura 5.4. As elipses serão usadas na etapa posterior como sementes da IFT.

Inicialmente, as imagens têm a escala reduzida para diminuir o tempo de processamento. Esta redução de escala é realizada somente para a localização de elipses. Ao final da etapa as elipses encontradas são ampliadas para a escala da imagem original. Os melhores resultados foram obtidos com a redução da imagem para 1/4 do tamanho original no *pipeline* de protozoários, para 1/8 do tamanho original no *pipeline* de ovos e para 1/10 do tamanho original no *pipeline* de larvas.

A imagem quantizada é limiarizada com um valor fixo. Como o processamento das amostras e a aquisição das imagens seguem um protocolo definido, não há diferença significativa de brilho e contraste entre as imagens, e assim um limiar fixo pode ser estabelecido. Após sucessivos testes, o melhor limiar foi estabelecido em 30 para o *pipeline* de protozoários e larvas, e em 25 para o *pipeline* de ovos. A técnica de limiarização automática de Otsu [39] foi testada, porém obteve desempenho inferior. Como trabalho futuro, será investigado uma forma de automatizar a seleção de limiar, para tornar o sistema robusto a problemas de descalibração do equipamento de aquisição de imagens.

A etapa de reconstrução superior, existente nos *pipelines* para protozoários e ovos de helmintos, é usada para corrigir falhas da limiarização e fechar regiões no interior dos

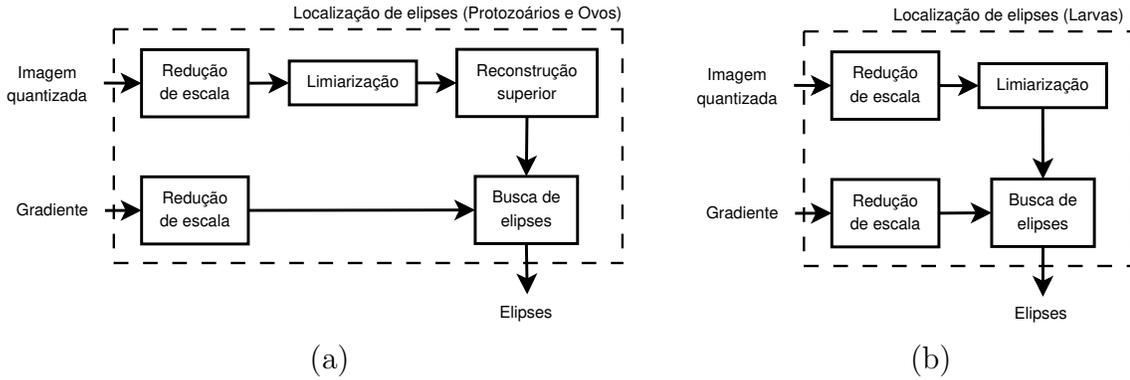


Figura 5.4: Detalhes da etapa de localização de elipses. (a) Etapa para protozoários e ovos de helmintos. (b) Etapa para larvas.

componentes. A figura 5.5c mostra a imagem 5.5a após a limiarização e reconstrução superior.

A imagem binária resultante é usada como máscara, e para cada pixel da imagem é encontrada a elipse que melhor se encaixa na máscara binária [34]. Para cada pixel p da máscara são traçadas m retas imaginárias com rotação de 0 a 180 graus, que passam pelo pixel. No experimento realizado, o melhor resultado foi obtido com $m = 24$. Para cada reta, é encontrado o ponto mais próximo de p que está fora da máscara. A coordenada relativa deste ponto em relação ao pixel $p = (x_p, y_p)$ no centro da elipse é representado por (u_i, v_i) . Seja um ângulo γ dado por

$$\gamma = \frac{\arctan \left[\frac{2S_{uv}}{S_v^2 - S_u^2} \right]}{2} \quad (5.2)$$

onde

$$S_{uv} = \sum_{i=1, \dots, m} u_i v_i \quad (5.3)$$

$$S_u^2 = \sum_{i=1, \dots, m} u_i^2 \quad (5.4)$$

$$S_v^2 = \sum_{i=1, \dots, m} v_i^2 \quad (5.5)$$

e as coordenadas (u_{r_i}, v_{r_i}) dos pontos rotacionados pelo ângulo γ calculadas por

$$u_{r_i} = u_i \cos \gamma - v_i \sin \gamma \quad (5.6)$$

$$v_{r_i} = u_i \sin \gamma + v_i \cos \gamma \quad (5.7)$$

Os eixos a e b da elipse com centro em (x_p, y_p) , e seu ângulo de rotação θ são encontrados através das equações

$$a = \sqrt{\frac{C \cdot C - D \cdot E}{C \cdot B - A \cdot E}} \quad (5.8)$$

$$b = \sqrt{\frac{C \cdot C - D \cdot E}{A \cdot C - D \cdot B}} \quad (5.9)$$

$$\theta = -\gamma \quad (5.10)$$

onde

$$A = \sum_{i=1, \dots, m} u_{r_i}^2 \quad (5.11)$$

$$B = \sum_{i=1, \dots, m} v_{r_i}^2 \quad (5.12)$$

$$C = \sum_{i=1, \dots, m} u_{r_i}^2 v_{r_i}^2 \quad (5.13)$$

$$D = \sum_{i=1, \dots, m} u_{r_i}^4 \quad (5.14)$$

$$E = \sum_{i=1, \dots, m} v_{r_i}^4 \quad (5.15)$$

O peso desta elipse é dado pela soma do gradiente dos pixels ao longo do contorno, dividido pelo perímetro (gradiente médio do contorno). Quanto maior o peso, mais elíptico é o componente coberto pela elipse. Para evitar a concentração de elipses em determinadas regiões da imagem, foi definido um critério de distância mínima entre os centros das elipses. Caso seja encontrada uma elipse com peso maior que outra já calculada, e com distância menor que a distância mínima definida, a nova elipse substitui a elipse antiga. Esta distância deve ser inferior à largura do menor parasito de cada grupo, para permitir a localização de uma elipse em cada parasito quando os parasitos estão localizados lado a lado. A distância mínima utilizada foi de $3,5\mu m$ para protozoários, $18,75\mu m$ para ovos e $14\mu m$ para larvas.

As medidas de comprimento e largura dos parasitos especificadas nas tabelas 4.1 e 4.2 são usadas para limitar o menor e maior tamanho possível das elipses, e assim reduzir o número de elipses encontradas. Apesar das larvas não terem a forma elíptica, a técnica encontra elipses em partes do parasito, sendo suficientes para o delineamento da larva inteira. A figura 5.5d ilustra as elipses encontradas pelo *pipeline* de ovos com a imagem original ao fundo.

4. **Watershed** - Usa as elipses de maior peso, com escala da imagem original, como marcadores da transformada de *watershed*, executada sobre a imagem de gradiente. Cada elipse é erodida e usada como semente interna (componente). Para os *pipelines* de protozoários e ovos de helmintos, a elipse também é dilatada e seu contorno é usado como semente externa (fundo da imagem). Para o *pipeline* de larvas, o negativo da imagem limiarizada, contendo o fundo de imagem, é erodido e usado como semente externa. Em todos os casos, os pixels da borda (*frame*) da imagem são marcados como semente externa. O uso da elipse dilatada como semente externa serve para eliminar as impurezas que encobrem parte dos parasitos. O gradiente fica fraco nas partes encobertas, e este marcador impede o vazamento do *watershed* por essas regiões. A figura 5.6 ilustra os marcadores obtidos através das elipses presentes na figura 5.5d. A figura 5.7 mostra os componentes delineados para os marcadores da figura 5.6. Esta etapa pode gerar componentes com poucas diferenças no delineamento, caso duas ou mais elipses fiquem em regiões diferentes do mesmo componente, e a eliminação de tais componentes é realizada após a classificação, na etapa de remoção de componentes semelhantes. As figuras 5.6e e 5.7e ilustram um exemplo onde a seleção da semente externa por dilatação da elipse pode gerar componentes parecidos com parasitos. O componente é eliminado na etapa de remoção de componentes semelhantes por ser um subconjunto do componente 5.7f.
5. **União de contornos** - Esta etapa é exclusiva do *pipeline* de larvas, executada após a transformada de *watershed*. A etapa verifica se o componente segmentado possui, além do contorno externo, um contorno interno. Caso possua, há uma região de fundo de imagem no interior do componente, e o contorno interno é unido ao contorno externo no ponto de menor distância entre eles. A figura 5.5e ilustra o resultado do *watershed* para a imagem 5.1a no *pipeline* para larvas, e a figura 5.5f mostra a união dos contornos interno e externo da larva. Esta etapa é importante para a extração correta de medidas das larvas, usadas posteriormente para a classificação.
6. **Remoção de componentes semelhantes** - Este pós-processamento é realizado após a classificação dos componentes. Todos os componentes são comparados entre si, e para componentes com superposição maior que 50%, é selecionado aquele com a maior pontuação do classificador. Os outros componentes com pontuação inferior são eliminados.

O delineamento gera máscaras binárias dos componentes segmentados, usadas na fase de reconhecimento.

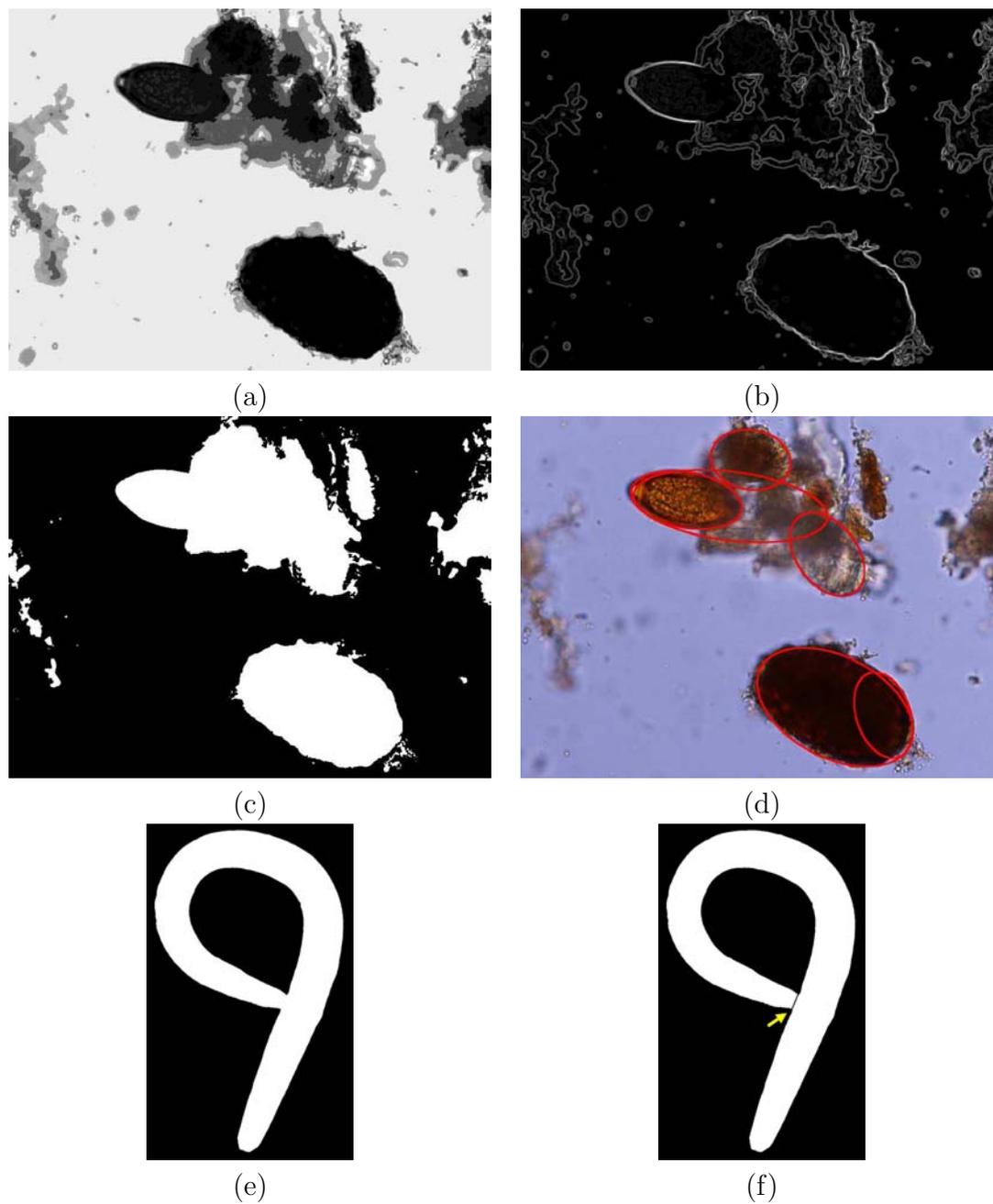


Figura 5.5: (a) Imagem 5.1e após a etapa de quantização. (b) Imagem **a** após a etapa de realce de borda. (c) Imagem **a** após a limiarização e reconstrução superior. (d) Ilustração das elipses encontradas pelo *pipeline* para ovos, com a imagem original ao fundo. (e) Resultado do *watershed* para a imagem 5.1a no *pipeline* para larvas. (f) Resultado da imagem e após a etapa de união de contornos. A seta indica o local de união dos contornos.

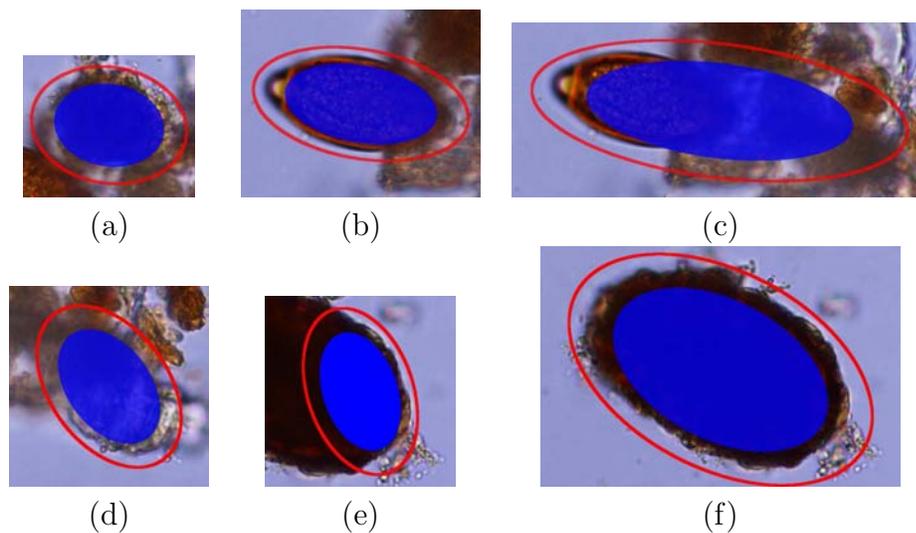


Figura 5.6: (a-f) Exemplos de marcadores para a transformada de *watershed*, obtidos através das elipses ilustradas na figura 5.5d. A elipse azul representa o marcador interno (componente), e a elipse vermelha representa o marcador externo (fundo da imagem). Na figura (b) o marcador está sobre um ovo de *Trichuris trichiura*, e na figura (f) o marcador está sobre um ovo de *Ascaris lumbricoides*.

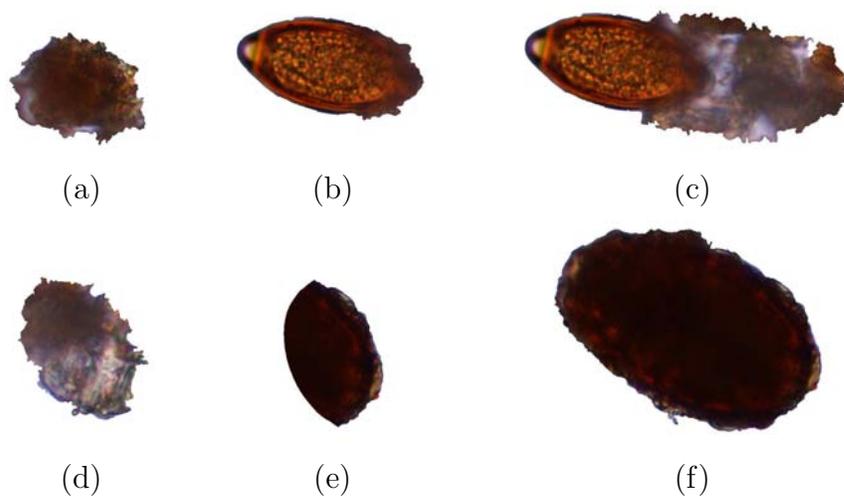


Figura 5.7: Exemplos de componentes delineados para os marcadores ilustrados na figura 5.6. Os componentes são usados para o reconhecimento. Todos os classificadores rotulam corretamente os componentes (b) e (f) como parasitos, e os outros componentes como impurezas.

5.2.2 Reconhecimento

A fase de reconhecimento extrai características relevantes dos componentes delineados, e usa essas características como entrada de um classificador. O classificador é responsável por rotular o componente como parasito ou não-parasito.

Para cada componente são extraídas 12 características:

- **Elipsidade** - É calculada a elipse sobre o componente, e a medida é a razão entre os pixels do componente que ficaram dentro da elipse e a área do componente. Como os protozoários e ovos de helmintos são circulares ou elípticos, e as impurezas são normalmente irregulares, esta característica indica um valor alto para parasitos (exceto larvas), e valor baixo para a maioria das impurezas.
- **Razão entre a maior e a menor distância geodésica** - A distância geodésica entre dois pontos é definida como o comprimento do caminho geodésico entre eles. Considerando todos os pares de pixels de localização oposta no contorno do componente, é calculada a razão entre a maior e menor distância geodésica. A característica diferencia componentes bem alongados, como larvas, de componentes circulares ou pouco alongados.
- **Variância das curvaturas ao longo do contorno** - A curvatura de um contorno $c(t) = (x(t), y(t))$ é definida para cada ponto t por

$$k(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'^2(t) + y'^2(t))^{3/2}} \quad (5.16)$$

onde $x'(t)$ e $y'(t)$ são a primeira derivada, e $x''(t)$ e $y''(t)$ são a segunda derivada de um ponto t do contorno. As derivadas podem ser obtidas com a transformada de Fourier e sua inversa, com

$$F(f^{(n)}) = (2\pi ik)^n F(f) \quad (5.17)$$

$$f^{(n)} = F^{-1}(F(f^{(n)})) \quad (5.18)$$

onde $F(f)$ é a transformada de Fourier de $c(t)$, e n é a ordem da derivada. Para evitar instabilidades no cálculo das curvaturas, o contorno discreto é suavizado com um filtro Gaussiano

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (5.19)$$

com média zero e desvio padrão σ .

Após o cálculo das curvaturas dos pontos do contorno, a variância σ^2 é encontrada com

$$\sigma^2 = \frac{1}{N_c - 1} \sum_{i=1}^{N_c} (k_i - \bar{k})^2 \quad (5.20)$$

onde N_c é o número de pontos do contorno, k_i é a curvatura do ponto i e \bar{k} é a média das curvaturas. Esta medida deve ser alta para impurezas, geralmente com contornos irregulares, e baixa para parasitos.

- **Variância das saliências ao longo do contorno** - Para cada ponto de contorno, sua saliência é definida como a área da maior zona de influência deste ponto, ou fora ou dentro do componente, em um diagrama de Voronoi, definido em torno do contorno [12]. As zonas de influência são calculadas através da IFT. Os valores de saliência são afetados pela forma, quando partes opostas do contorno estão próximas e impedem uma extensão maior da área de influência dos pontos nesses trechos. Esta informação sobre a geometria do contorno não é representada pelas curvaturas.

A variância é calculada com as saliências ao longo do contorno, e tende a ser mais baixa para parasitos do que para impurezas.

- **Textura do vermelho** - O descritor é usado para diferenciar componentes através da variação de textura na banda vermelha da imagem original colorida. Como o corante utilizado deixa os parasitos com um tom castanho, as estruturas internas têm melhor definição na banda vermelha.

A imagem é filtrada com uma operação de fechamento por área [33], que preenche regiões conexas escuras com área menor que um valor λ especificado. Nos testes realizados, o melhor resultado foi obtido com $\lambda = (\text{área do componente})/4$. Com isso, o filtro elimina detalhes no interior do componente sem modificar a forma, simplificando a textura. A imagem inicial é subtraída da imagem filtrada, resultando em uma imagem de resíduos. Para componentes com grande variação de textura, a imagem resultante deve conter uma grande quantidade de resíduos. No caso de componentes lisos ou com baixa variação de textura, a imagem filtrada deve ficar próxima da imagem original e o processo deve gerar uma imagem com poucos resíduos. Ao final do processamento, é calculada a variância com os valores dos resíduos. Geralmente impurezas são lisas ou possuem menor variação de textura do que os parasitos.

- **Média de vermelho** - O descritor calcula a média dos valores dos pixels do componente na banda vermelha da imagem original colorida. A medida visa diferenciar componentes com média próxima dos parasitos, com tom castanho, dos demais componentes.
- **Razão entre o número de mínimos regionais do gradiente e a área do componente** - Usa a IFT para encontrar os mínimos regionais do componente, no gradiente da imagem original (não quantizada). A medida é a razão entre o número

de mínimos regionais encontrado e a área do componente. Com o mesmo objetivo do descritor de textura do vermelho, esta medida tenta capturar variações na textura dos componentes, diferenciando componentes lisos (impurezas) de componentes com maior variação de textura.

- **Perímetro** - Calcula o perímetro do componente. Esta medida é usada para diferenciar componentes com perímetro fora do padrão dos parasitos.
- **Área** - Calcula a quantidade de pixels do componente. A medida é usada para diferenciar componentes com área fora do padrão dos parasitos.
- **Momentos** - São calculados o segundo, terceiro e quarto momentos do histograma do componente quantizado em 512 níveis de cinza. O valor quantizado de um dado pixel p é obtido pela equação

$$I_q(p) = \left\lfloor \frac{I_R(p)}{32} \right\rfloor + 8 \left\lfloor \frac{I_G(p)}{32} \right\rfloor + 64 \left\lfloor \frac{I_B(p)}{32} \right\rfloor \quad (5.21)$$

onde $I_R(p)$, $I_G(p)$ e $I_B(p)$ são os valores das bandas vermelha, verde e azul do pixel p na imagem original colorida.

O segundo momento mede a variância σ^2 do histograma, calculada através da equação 5.20. O terceiro momento (*skewness*) mede a simetria do histograma, e é calculado por

$$Skew = \frac{1}{N_h} \sum_{i=1}^{N_h} \left[\frac{h_i - \bar{h}}{\sigma} \right]^3 \quad (5.22)$$

onde $N_h = 512$ é o tamanho do histograma \mathbf{h} , \bar{h} é a média do histograma e σ é o desvio padrão. O quarto momento (*kurtosis*) avalia a concentração em torno da média, e é calculado por

$$Kurt = \left\{ \frac{1}{N_h} \sum_{i=1}^{N_h} \left[\frac{h_i - \bar{h}}{\sigma} \right]^4 \right\} - 3 \quad (5.23)$$

Os vetores de características extraídos dos componentes são usados como entrada dos classificadores, que rotulam os componentes como parasitos ou impurezas.

A classificação foi avaliada com cinco classificadores: Rede Neural MLP (Seção 3.1) com *Bagging* (Seção 3.4.1), Rede Neural MLP com *Adaboost* (Seção 3.4.2), SVM (Seção 3.2), SVM com *Adaboost* e OPF (Seção 3.3). Os cinco classificadores obtiveram desempenho acima de 90%, com baixa quantidade de falsos positivos. O melhor desempenho foi alcançado com o classificador OPF, que detectou corretamente 97,36% dos parasitos, com 4,73% de falsos positivos.

Capítulo 6

Experimentos e resultados

As 407 imagens da base (Seção 4) são utilizadas como entrada do delineamento dos três *pipelines*, que extraem 2.774 componentes, contendo parasitos e impurezas. Neste ponto é possível avaliar a qualidade do delineamento automático em relação à segmentação manual dos 455 parasitos, e os resultados desta avaliação são apresentados na seção 6.1. Os componentes delineados são usados como entrada da fase de reconhecimento, que calcula os vetores de características dos componentes e usa estes vetores para a classificação dos componentes. Os experimentos foram realizados com cinco classificadores: Rede Neural MLP com *Bagging*, Rede Neural MLP com *Adaboost*, SVM, SVM com *Adaboost* e OPF. O treinamento dos classificadores é detalhado na seção 6.2. Na etapa de classificação, os componentes são rotulados como parasitos ou impurezas, e neste ponto é possível avaliar o desempenho de cada classificador em relação ao total de componentes classificados (parasitos e impurezas) e ao total de parasitos detectados. Os resultados desta avaliação são apresentados na seção 6.3.

Em um computador Pentium4 com 3GHz, a segmentação completa (delineamento + reconhecimento) de uma imagem nos três *pipelines* demora em média 9 segundos.

6.1 Resultados do delineamento

Estes resultados refletem a qualidade da segmentação com relação à tarefa de delineamento. Cada componente delineado é comparado pixel a pixel com o parasito segmentado manualmente nesta imagem. Se o componente for uma impureza, então não haverá superposição com o parasito. Se o componente for o parasito, então as diferenças na superposição estarão associadas ao erro no delineamento. Este erro é medido como a soma de pixels falsos positivos e falsos negativos. Pixels falsos positivos são os pixels do componente que não pertencem ao parasito, e pixels falsos negativos são pixels do parasito que não pertencem ao componente. Estas medidas são normalizadas em duas taxas: *FN*

(falsos negativos) e FP (falsos positivos). A taxa FN é a razão entre o número de pixels falsos negativos e a área do parasito, e a taxa FP é a razão entre o número de pixels falsos positivos e a área do componente. Componentes que não possuem superposição com o parasito apresentam $FP+FN=2$, e componentes em superposição com o parasito apresentam $0 \leq FP+FN < 2$.

A tabela 6.1 indica o número total de componentes delineados para cada *pipeline*, e o número de componentes com taxa $FP+FN$ abaixo de 2, que podem ser parasitos com diferentes níveis de falhas no delineamento. Estes números incluem componentes com poucas diferenças no delineamento, pois a eliminação de tais componentes é realizada somente após a classificação. A diferença entre estes valores reflete a quantidade de impurezas nas imagens.

A tabela 6.2 indica o erro máximo $FP+FN$ observado, onde o componente delineado ainda preserva as características principais do parasito que permitem a identificação da espécie; o número de componentes com erro igual ou abaixo do limite; e o percentual de acerto do delineamento com relação ao número de parasitos da base. Com o objetivo de avaliar a quantidade de parasitos delineados corretamente, os componentes repetidos não foram contabilizados, sendo contado aquele com menor erro $FP+FN$.

Os resultados indicam que 98,24% dos 455 parasitos são delineados com sucesso, de acordo com o critério de erro máximo permitido.

<i>Pipeline</i>	Nº de componentes	Nº de componentes com $FP+FN < 2$
Protozoários	1.463	610
Ovos	870	506
Larvas	441	266
Total	2.774	1.382

Tabela 6.1: Total de componentes delineados e o número de componentes com erro abaixo de 2.

6.2 Treinamento dos classificadores

O treinamento dos classificadores é realizado com a base de projeto (Seção 4.3), separada em três grupos: protozoários, com 476 parasitos e 1579 não-parasitos (impurezas e parasitos com falhas graves no delineamento) de tamanho pequeno; ovos de helmintos, com 406 parasitos e 737 não-parasitos de tamanho grande; e larvas, com 22 parasitos e 212 não-parasitos com forma alongada. Cada grupo é usado para o treinamento dos

<i>Pipeline</i>	Erro máximo	Nº de comp. com $FP+FN \leq$ Erro máximo	% acerto
Protozoários	0,22	234	97,50%
Ovos	0,35	202	99,02%
Larvas	0,40	11	100,00%
Total	—	447	98,24%

Tabela 6.2: Erro máximo permitido para identificação das espécies, número de componentes igual ou abaixo do erro máximo e a taxa de acerto do delineamento.

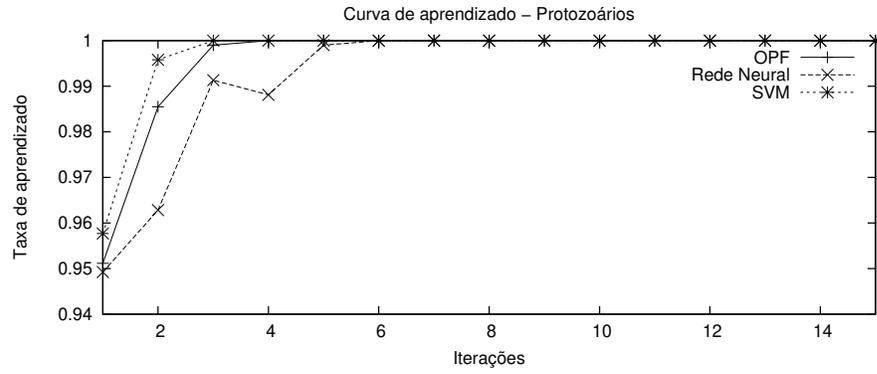
classificadores de um *pipeline*. A base foi separada em 50% para treinamento e 50% para avaliação do treinamento.

Os classificadores Rede Neural MLP, SVM e OPF foram treinados segundo o algoritmo de aprendizado apresentado na Seção 3.3.1, com 15 iterações. As figuras 6.1a-c mostram as curvas de aprendizado com a comparação de acurácia dos classificadores, para imagens de protozoários, ovos e larvas. O número de amostras de cada classe, no conjunto de treinamento, foi limitado em 50% do total de amostras da classe. Os gráficos mostram que os classificadores OPF e SVM possuem curvas de aprendizado semelhantes, e aprendem rapidamente, chegando ao valor máximo em poucas iterações. Observa-se também a instabilidade da Rede Neural, perdendo acurácia após alcançar o valor máximo. Os três classificadores alcançaram um valor alto de acurácia para o conjunto de avaliação do treinamento, porém, os classificadores SVM e Rede Neural não garantem o mesmo nível de acurácia para amostras do conjunto de treinamento, o que pode levar a um desempenho abaixo do esperado.

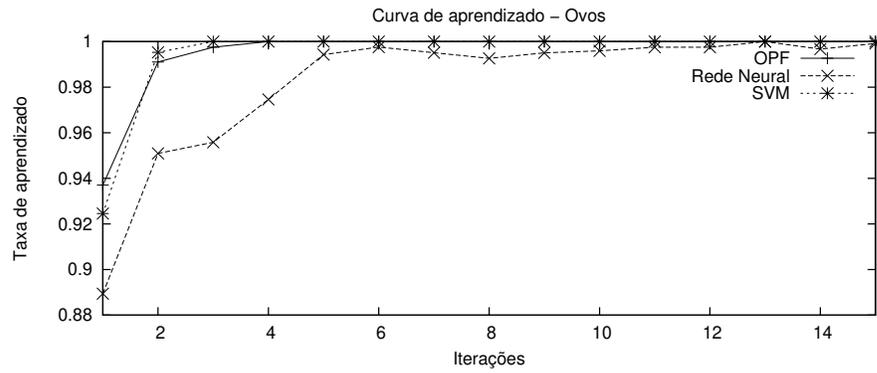
O classificador Rede Neural apresentou desempenho inferior aos outros classificadores, e foi usado com as técnicas para coleção de classificadores *Bagging* e *Adaboost* para obter melhor desempenho. As redes foram configuradas com 12 nós na camada de entrada (número de características), 18 nós na camada escondida e 2 nós na camada de saída (número de classes). As redes projetadas com número de nós acima ou abaixo de 18 na camada escondida apresentaram desempenho inferior. Para a coleção treinada com *Bagging*, o melhor resultado foi obtido com 9 instâncias do classificador, e para a coleção treinada com *Adaboost* o melhor resultado foi obtido com 15 instâncias.

O classificador SVM foi testado também com *Adaboost*, com 15 instâncias do classificador, e não houve melhoria de desempenho com a técnica *Bagging*. O treinamento gerou 292 vetores de suporte para protozoários, 403 vetores de suporte para ovos de helmintos e 14 vetores de suporte para larvas.

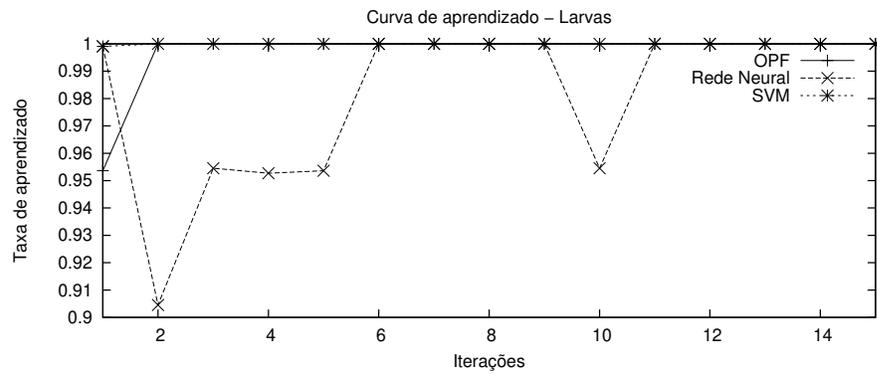
O classificador OPF não obteve melhoria de desempenho com as técnicas para coleção de classificadores, e foi usado sem combinação. O treinamento do classificador gerou 126



(a)



(b)



(c)

Figura 6.1: (a) Curva de aprendizado para a base de protozoários. (b) Curva de aprendizado para a base de ovos. (c) Curva de aprendizado para a base de larvas.

protótipos para protozoários, 141 protótipos para ovos de helmintos e 9 protótipos para larvas.

No total, foram avaliados 5 classificadores em cada *pipeline*: Rede Neural com *Bagging*, Rede Neural com *Adaboost*, OPF, SVM e SVM com *Adaboost*.

6.3 Resultados após o reconhecimento

Estes resultados refletem a qualidade do delineamento com relação ao reconhecimento (i.e., desempenho dos classificadores). Os resultados verdadeiros positivos (*VP*) e verdadeiros negativos (*VN*) são considerados classificações corretas. Verdadeiros positivos são parasitos corretamente classificados como parasitos, e verdadeiros negativos são impurezas corretamente classificadas como impurezas. Falsos positivos (*FP*), neste contexto, são impurezas incorretamente classificadas como parasitos.

A tabela 6.3 mostra o desempenho dos classificadores em relação ao total de componentes delineados. A taxa de acerto é dada pela razão entre a soma $VP+VN$ e o número de componentes.

A tabela 6.4 mostra o desempenho dos classificadores em relação ao total de parasitos, após a remoção de componentes semelhantes (Seção 5.2.1). Neste caso, a taxa de acerto é calculada através da razão entre *VP* e o total de parasitos.

As tabelas 6.5-6.14 apresentam as comparações de acertos e erros entre todos os pares de classificadores usados, considerando todos os componentes delineados. Para a avaliação nós usamos o teste de McNemar [30], através da equação

$$\chi^2 = \frac{(|N_{01} - N_{10}| - 1)^2}{N_{01} + N_{10}} \quad (6.1)$$

onde N_{01} é o número de vezes que o primeiro classificador erra e o segundo classificador acerta, e N_{10} é o número de vezes que o primeiro classificador acerta e o segundo classificador erra. A hipótese nula é que não há diferença de acurácia entre os classificadores. O resultado da equação 6.1 é comparado com o valor tabulado de χ^2 . Para um nível de significância 0,05, se $\chi^2 > 3,841$ nós rejeitamos a hipótese nula e aceitamos que os classificadores possuem desempenhos diferentes. A tabela 6.15 mostra o valor calculado de χ^2 para todos os pares de classificadores.

Conforme a tabela 6.3, o classificador SVM com *Adaboost* detectou mais protozoários que o classificador SVM simples, porém com uma quantidade maior de falsos positivos. Por outro lado, o uso da técnica de coleção de classificadores fez diminuir o número de ovos detectados. O teste de McNemar indica que os dois classificadores possuem desempenhos semelhantes, e não é possível determinar o melhor classificador. Segundo o teste, rejeitamos a hipótese nula para os pares OPF e Rede Neural com *Bagging*, OPF

Classificador	Nº de componentes	VP	VN	FP	% acerto
Rede Neural com <i>Bagging</i>	2.774	425	2.275	33	97,33%
Protozoários	1.463	223	1.214	15	98,22%
Ovos	870	182	641	18	94,60%
Larvas	441	20	420	0	99,77%
Rede Neural com <i>Adaboost</i>	2.774	435	2.277	31	97,77%
Protozoários	1.463	225	1.208	21	97,95%
Ovos	870	189	649	10	96,32%
Larvas	441	21	420	0	100,00%
OPF	2.774	456	2.275	33	98,45%
Protozoários	1.463	234	1.212	17	98,84%
Ovos	870	202	643	16	97,13%
Larvas	441	20	420	0	99,77%
SVM	2.774	439	2.287	21	98,27%
Protozoários	1.463	222	1.216	13	98,29%
Ovos	870	197	655	4	97,93%
Larvas	441	20	416	4	98,87%
SVM com <i>Adaboost</i>	2.774	434	2.282	27	97,87%
Protozoários	1.463	227	1.209	20	98,15%
Ovos	870	187	656	3	96,90%
Larvas	441	20	416	4	98,87%

Tabela 6.3: Total de componentes que passam pelos classificadores, valores *VP*, *VN* e *FP*, e percentual de acertos na classificação.

e Rede Neural com *Adaboost*, e SVM e Rede Neural com *Bagging*. Para todos os outros pares o teste indica que não há diferença significativa na acurácia dos classificadores. Portanto, o classificador OPF obteve desempenho superior aos classificadores Rede Neural com *Bagging* e Rede Neural com *Adaboost*, e desempenho semelhante aos classificadores SVM e SVM com *Adaboost*. Considerando a fase de treinamento, o classificador OPF é superior ao classificador SVM pois alcança desempenho semelhante com um número menor de protótipos (276 protótipos para o classificador OPF e 709 vetores de suporte para o classificador SVM), ou seja, o classificador OPF possui uma capacidade de aprendizado (Equação 3.21) superior ao classificador SVM. Ao considerar somente os parasitos da base de imagens detectados corretamente, o melhor resultado foi obtido com o classificador OPF, separando 97,36% dos parasitos, com 4,73% de falsos positivos.

Classificador	Nº de parasitos	VP	FP	% acerto
Rede Neural com <i>Bagging</i>	455	410	23	90,11%
Protozoários	240	223	13	92,92%
Ovos	204	176	10	86,27%
Larvas	11	11	0	100,00%
Rede Neural com <i>Adaboost</i>	455	421	24	92,53%
Protozoários	240	225	19	93,75%
Ovos	204	185	5	90,67%
Larvas	11	11	0	100,00%
OPF	455	443	22	97,36%
Protozoários	240	234	13	97,50%
Ovos	204	198	9	97,06%
Larvas	11	11	0	100,00%
SVM	455	429	18	94,29%
Protozoários	240	222	12	92,50%
Ovos	204	196	3	96,08%
Larvas	11	11	3	100,00%
SVM com <i>Adaboost</i>	455	424	22	93,19%
Protozoários	240	227	18	94,58%
Ovos	204	186	1	91,18%
Larvas	11	11	3	100,00%

Tabela 6.4: Número de parasitos da base, número de parasitos detectados corretamente (*VP*), número de não-parasitos detectados incorretamente como parasitos (*FP*), e percentual de acertos considerando a quantidade *VP*.

	RN c/ <i>Adaboost</i> correto	RN c/ <i>Adaboost</i> errado
RN c/ <i>Bagging</i> correto	2674	26
RN c/ <i>Bagging</i> errado	38	36

Tabela 6.5: Número de acertos e erros para o par de classificadores Rede Neural com *Bagging* e Rede Neural com *Adaboost*.

	OPF correto	OPF errado
RN c/ <i>Bagging</i> correto	2673	27
RN c/ <i>Bagging</i> errado	58	16

Tabela 6.6: Número de acertos e erros para o par de classificadores Rede Neural com *Bagging* e OPF.

	SVM correto	SVM errado
RN c/ <i>Bagging</i> correto	2675	25
RN c/ <i>Bagging</i> errado	51	23

Tabela 6.7: Número de acertos e erros para o par de classificadores Rede Neural com *Bagging* e SVM.

	SVM c/ <i>Adaboost</i> correto	SVM c/ <i>Adaboost</i> errado
RN c/ <i>Bagging</i> correto	2669	31
RN c/ <i>Bagging</i> errado	46	28

Tabela 6.8: Número de acertos e erros para o par de classificadores Rede Neural com *Bagging* e SVM com *Adaboost*.

	OPF correto	OPF errado
RN c/ <i>Adaboost</i> correto	2685	27
RN c/ <i>Adaboost</i> errado	46	16

Tabela 6.9: Número de acertos e erros para o par de classificadores Rede Neural com *Adaboost* e OPF.

	SVM correto	SVM errado
RN c/ <i>Adaboost</i> correto	2686	26
RN c/ <i>Adaboost</i> errado	40	22

Tabela 6.10: Número de acertos e erros para o par de classificadores Rede Neural com *Adaboost* e SVM.

	SVM <i>c/ Adaboost</i> correto	SVM <i>c/ Adaboost</i> errado
RN <i>c/ Adaboost</i> correto	2678	34
RN <i>c/ Adaboost</i> errado	37	25

Tabela 6.11: Número de acertos e erros para o par de classificadores Rede Neural com *Adaboost* e SVM com *Adaboost*.

	SVM correto	SVM errado
OPF correto	2703	28
OPF errado	23	20

Tabela 6.12: Número de acertos e erros para o par de classificadores OPF e SVM.

	SVM <i>c/ Adaboost</i> correto	SVM <i>c/ Adaboost</i> errado
OPF correto	2692	39
OPF errado	23	20

Tabela 6.13: Número de acertos e erros para o par de classificadores OPF e SVM com *Adaboost*.

	SVM <i>c/ Adaboost</i> correto	SVM <i>c/ Adaboost</i> errado
SVM correto	2698	28
SVM errado	17	31

Tabela 6.14: Número de acertos e erros para o par de classificadores SVM e SVM com *Adaboost*.

χ^2	RN <i>Bagging</i>	RN <i>Adaboost</i>	OPF	SVM	SVM <i>Adaboost</i>
RN <i>Bagging</i>	-	1,891	10,588	8,224	2,545
RN <i>Adaboost</i>	1,891	-	4,438	2,561	0,056
OPF	10,588	4,438	-	0,314	3,629
SVM	8,224	2,561	0,314	-	2,222
SVM <i>Adaboost</i>	2,545	0,056	3,629	2,222	-

Tabela 6.15: Valores calculados de χ^2 para os pares de classificadores. Em destaque estão os pares com a hipótese nula rejeitada (os classificadores possuem desempenhos diferentes).

Capítulo 7

Conclusões

Neste trabalho, desenvolvemos um método de segmentação para separar automaticamente os parasitos intestinais do Homem de outras estruturas presentes nas imagens de amostras fecais.

Os parasitos foram divididos em três grupos com características semelhantes: protozoários, ovos de helmintos e larvas de helmintos. Para cada grupo desenvolvemos um sistema especialista (*pipeline*), e cada imagem de amostra fecal é processada nos três *pipelines*.

A técnica proposta explora a característica de forma elíptica dos parasitos para a localização aproximada dos componentes, e usa a segmentação por *watershed* para o delineamento preciso. Com esta estratégia, nós alcançamos uma sensibilidade de 97,36% na detecção dos parasitos, com 4,73% de falsos positivos. O resultado é superior ao trabalho apresentado por Yang et al. (2001) [45], que alcançou 84% de acerto na detecção de ovos de helmintos e 10% de falsos positivos em 87 parasitos. O resultado também é relevante quando comparado com a baixa sensibilidade diagnóstica dos métodos tradicionais (48,3% a 75,9%) [29], e viabiliza o desenvolvimento de um sistema para diagnóstico, que permite a análise automática das imagens e separação dos parasitos detectados para o diagnóstico final por um especialista.

Uma das razões para a baixa sensibilidade diagnóstica dos métodos tradicionais é que o microscopista analisa a lâmina com objetiva de aumento de 100 vezes, e só passa para aumento de 400 vezes quando há suspeita de presença de parasitos. Em razão da grande diferença de tamanho entre protozoários e helmintos, muitos protozoários podem passar despercebidos na análise com aumento de 100 vezes. Além disso, somente uma pequena quantidade do sedimento fecal é separado para análise (aproximadamente 1/12), o que leva também a aumentar a possibilidade de erro. Considerando estes aspectos, todas as imagens utilizadas no projeto foram obtidas com aumento de 400 vezes, e o sistema de aquisição automática de imagens permitirá a leitura de todo o sedimento fecal. Desta

forma, o sistema permitirá a análise de uma quantidade de imagens superior à capacidade humana, com taxa de erro menor que a obtida com técnicas tradicionais.

Os *pipelines* podem ser executados em paralelo, com o uso de computadores com múltiplos processadores ou múltiplos núcleos (*multi-core*), que estão se tornando padrão da indústria para aplicações comerciais e domésticas. Em um computador multiprocessado, o tempo de processamento pode ficar próximo de 3 segundos, com a execução paralela de um *pipeline* em cada processador.

Nós desenvolvemos também um novo classificador de padrões, que interpreta o conjunto de treinamento como um grafo completo e constrói uma floresta de caminhos ótimos (OPF) sobre o grafo. Este classificador alcançou taxa de detecção de parasitos superior aos outros classificadores testados, com o uso de uma quantidade pequena de nós protótipos.

7.1 **Trabalhos futuros**

7.1.1 **Classificação das espécies parasitárias**

A próxima etapa consiste na pesquisa de descritores adequados para a classificação das espécies parasitárias, e formas de combinação de descritores com o objetivo de aumentar o desempenho do classificador.

Na pesquisa realizada, cada descritor extrai do componente delineado um número escalar, que agrupado forma um vetor de características do componente. Há descritores mais complexos e não abordados na pesquisa que extraem uma assinatura (vetor de características), com mais informações do componente, e são acompanhados por uma medida de similaridade específica, que calcula a similaridade entre dois componentes como uma função da distância entre os vetores de características correspondentes. Como exemplos estão o *Beam Angle Statistics* (BAS) [2], um descritor de forma baseado em curvatura que é invariante a translação, rotação e escala, e o *Border/Interior Pixel Classification* (BIC) [43], um descritor que realiza a quantização da imagem e extrai histogramas para pixels na borda do componente e pixels no interior do componente. Como possuem função de distância específica para cada descritor, não é possível agrupar os vetores de características de todos os descritores e aplicar uma função de distância única.

Uma pesquisa recente é o uso do classificador OPF com programação genética para a combinação de descritores. A programação genética é usada para a criação de um descritor composto [10], que combina as medidas de similaridade de todos os descritores utilizados e gera um valor de distância, usado como peso de aresta no classificador OPF. A figura 7.1a ilustra o descritor composto. O módulo de programação genética é treinado para se obter uma árvore de expressões matemáticas que maximiza a distância para componentes de classes diferentes e minimiza esse valor para componentes de mesma classe. A figura 7.1b

ilustra uma árvore de expressões matemáticas, onde as folhas são os valores de similaridade dos descritores.

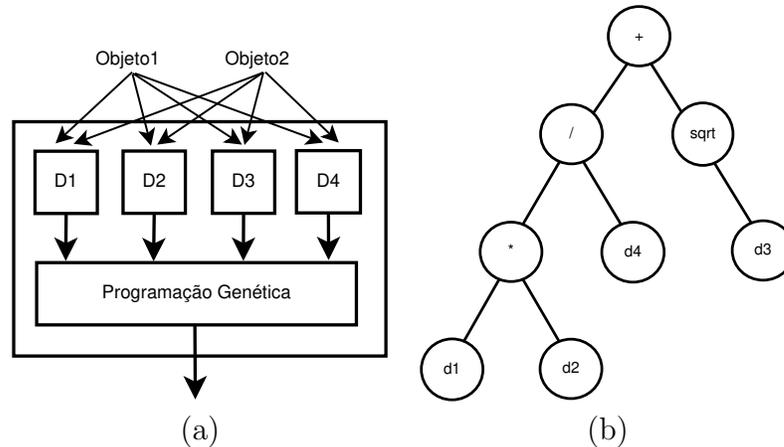


Figura 7.1: (a) Exemplo de descritor composto - As medidas de similaridade dos descritores D1 a D4 são combinadas através do módulo de programação genética para se obter uma única medida de saída. (b) Exemplo de árvore de expressões matemáticas gerada por programação genética. As folhas são os valores de similaridade dos descritores.

7.1.2 Desenvolvimento de software para diagnóstico

Consiste no desenvolvimento de um software com interface gráfica que combina a técnica de segmentação com a classificação por espécie parasitária, para auxiliar na operação do sistema e visualização dos resultados.

7.1.3 Avaliação do sistema com mais imagens

Será necessário avaliar o sistema com um número maior de imagens, que estão sendo obtidas pelo aluno da Biologia.

7.1.4 Automação do sistema

Uma próxima atividade consiste na adaptação do software de diagnóstico a um equipamento de aquisição automática de imagens, que permitirá a captura de imagens de todo o campo visual da lâmina e a leitura de todo o sedimento fecal.

As vantagens do sistema de diagnóstico automático em relação aos procedimentos adotados atualmente nos laboratórios de análises clínicas são:

- Maior produtividade;
- Maior sensibilidade diagnóstica;
- Menor despesa com mão-de-obra;
- Redução de espaço físico no laboratório de parasitologia;
- Redução no uso e lavagens de utensílios;
- Rapidez, praticidade e segurança diagnóstica;
- Diminuição do risco de contato e contaminação do profissional de laboratório;
- Vantagens de custo-benefício a curto e médio prazos;
- Descarte de material analisado em recipiente apropriado para líquidos tóxicos, seguindo normas de biossegurança;
- Inovação na liberação dos resultados através de imagens;
- Resgate da credibilidade no diagnóstico de parasitoses intestinais por parte da classe médica, evitando o uso empírico, indiscriminado e ineficaz de medicamentos.

Referências Bibliográficas

- [1] Mpeg-7: The generic multimedia content description standard, part 1. *IEEE Multi-Media*, 09(2):78–87, 2002.
- [2] Nafiz Arica and Fatos T. Yarman Vural. Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recogn. Lett.*, 24(9-10):1627–1639, 2003.
- [3] Y. Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 105–112, 2001.
- [4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [5] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [6] T. Cootes, G. Edwards, and C.J.Taylor. Active appearance models. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 484–498, 1998.
- [7] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [8] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. MIT, Cambridge, MA, USA, 1990.
- [9] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14(3):326–334, Jun 1965.
- [10] R. da S. Torres, A. X. Falcão, M. A. Goncalves, B. Zhang, W. Fan, E. A. Fox, and P. Calado. A New Framework to Combine Descriptors for Content-based Image Retrieval. In *Proceedings of the Fourteenth Conference on Information and Knowledge Management (CIKM05)*, pages 335–336, Bremen, Germany, November 2005. ACM Press.

- [11] R. da S. Torres, A.X. Falcão, and L.F. Costa. Shape description by image foresting transform. In *14th Intl. Conf. on Digital Signal Processing*, pages 1089–1092, Santorini, Greece, Jul 2002.
- [12] R. da S. Torres, A.X. Falcão, and L.F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004.
- [13] A. Dauschies, S. Imarom, and W. Bollwahn. Differentiation of porcine eimeria spp. by morphologic algorithms. *Veterinary Parasitology*, 81(3):201–210, 1999.
- [14] G.A. de Carli. *Parasitologia Clínica: Seleção de Métodos e Técnicas de Laboratório para o Diagnóstico das Parasitoses Humanas*. Atheneu, São Paulo, 2nd edition, 2001.
- [15] R.B. Dial. Shortest-path forest with topological ordering. *Comm. of the ACM*, 12(11):632–633, Nov 1969.
- [16] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [17] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.
- [18] A.X. Falcão and F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 23(9):1100–1108, 2004.
- [19] A.X. Falcão, L.F. Costa, and B.S. da Cunha. Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, 2002.
- [20] A.X. Falcão, B.S. da Cunha, and R.A. Lotufo. Design of connected operators using the image foresting transform. In *Proc. of SPIE on Medical Imaging*, volume 4322, pages 468–479, Feb 2001.
- [21] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.
- [22] A.X. Falcão and J.K. Udupa. A 3D generalization of user-steered live wire segmentation. *Medical Imaging Analysis*, 4(4):389–402, Dec 2000.
- [23] A.X. Falcão, J.K. Udupa, and F.K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, 2000.

- [24] A.X. Falcão, J.K. Udupa, S. Samarasekera, S. Sharma, B.E. Hirsch, and R.A. Lotufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):233–260, Jul 1998.
- [25] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
- [26] J.F. Gomes. Avaliação de um novo kit (*TF-Test*) nacional destinado ao diagnóstico de enteroparasitoses em amostras fecais. Master’s thesis, Universidade Estadual de Campinas - Unicamp, Campinas, SP, 2004.
- [27] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.
- [28] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, Upper Saddle River, NJ, USA, July 1998.
- [29] Gomes J.F., Hoshino-Shimizu S., Dias L.C., Araujo A.J., Castilho V.L., and Neves F.A. Evaluation of a novel kit (tf-test) for the diagnosis of intestinal parasitic infections. *Journal of clinical laboratory analysis*, 18(2):132–138, 2004.
- [30] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, Hoboken, NJ, USA, July 2004.
- [31] R.A. Lotufo and A.X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer, Jun 2000.
- [32] R.A. Lotufo, A.X. Falcão, and F.A. Zampirolli. Fast Euclidean distance transform using a graph-search algorithm. In *Proc. of the XIII Brazilian Symp. on Computer Graphics and Image Processing*, pages 269–275. IEEE, Oct 2000.
- [33] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):484–494, Apr 2002.
- [34] Paulo A. V. Miranda, Ricardo da S. Torres, and Alexandre X. Falcão. Tsd: A shape descriptor based on a distribution of tensor scale local orientation. In *SIBGRAPI '05: Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 139–146, Washington, DC, USA, 2005. IEEE Computer Society.

- [35] Thomas Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), New York, October 1997.
- [36] E.F. Moore. The shortest path through a maze. In *Proc. Intl. Symp. on the Theory of Switching*, pages 285–292. Harvard University, Apr 1959.
- [37] E.N. Mortensen and W.A. Barrett. Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384, 1998.
- [38] D.P. Neves, A.L. Melo, O. Genaro, and M.P. Linardi. *Parasitologia Humana*. Atheneu, São Paulo, tenth edition, 2001.
- [39] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9(1):62–66, March 1979.
- [40] João P. Papa, Celso T. N. Suzuki, Alexandre X. Falcão, and Nelson D. A. Mascarenhas. A new pattern classifier based on optimum path forest. *Submitted to IEEE International Conference on Image Processing*, 2007.
- [41] L. Rey. *Bases da Parasitologia Médica*. Guanabara Koogan, Rio de Janeiro, 2000.
- [42] C. Sommer. Quantitative characterization of texture used for identification of eggs of bovine parasitic nematodes. *Journal of Helminthology*, pages 72(2):179–182, 1998.
- [43] Renato O. Stehling, Mario A. Nascimento, and Alexandre X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002. ACM Press.
- [44] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Intl Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518, 2001.
- [45] Y.S. Yang, D.K. Park, and H.C. Kim. Automatic identification of human helminth eggs on microscopic fecal specimens using digital image processing and an artificial neural network. *IEEE Transactions on Biomedical Engineering*, pages 48(6):718–730, Jun 2001.
- [46] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. on Computers*, C-20(1):68–86, Jan. 1971.