

ALLISSON DANTAS DE OLIVEIRA

MalariaSystem: uma ferramenta para
diagnóstico automático de malária em
dispositivos móveis

RECIFE-PE
JULHO - 2014



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

MalariaSystem: uma ferramenta para diagnóstico automático de malária em dispositivos móveis

Dissertação de mestrado apresentada ao Curso de Pós-Graduação em Informática Aplicada da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Informática Aplicada.

Área de Concentração: Modelagem Computacional
Orientador: Jones Oliveira de Albuquerque

RECIFE-PE
JULHO - 2014

Ficha catalográfica

O48m

Oliveira, Allisson Dantas de

MalariaSystem: uma ferramenta para diagnóstico automático de malária em dispositivos móveis / Allisson Dantas de Oliveira. – Recife, 2014.

88 f. : il.

Orientador: Jones Oliveira de Albuquerque.

Dissertação (Mestrado em Informática Aplicada) – Universidade Federal Rural de Pernambuco, Departamento de Estatística e Informática, Recife, 2014.

Inclui referências e apêndice(s).

1. Inteligência artificial 2. Malária 3. Processamento de imagens 4. Diagnóstico 5. Informática I. Albuquerque, Jones Oliveira de, orientador II. Título

CDD 004

À
minha querida mãe
meu querido pai

Agradecimentos

Agradeço primeiramente a Deus por ter proporcionando esse momento tão especial na minha vida e ter colocado as pessoas certas no meu árduo caminho.

Aos meus pais, Aluizio Dantas e Josefa Valentina, meu infinito respeito e admiração ambos estiveram sempre ao meu lado nas decisões mais difíceis. À compreensão pela minha ausência em certos momentos familiares. À minha família e meus irmãos. Aos meus amigos(as) Ricardo Tavares, Samuel Luna, Dalton Araujo, dentre outros que me apoiaram neste projeto. A minha querida Dominique Campelo.

Ao Professor e Orientador Dr^o Jones Oliveira de Albuquerque, por ter confiado e acreditado no meu trabalho, que sempre se fez disponível a contribuir nesse projeto, ao Coorientador André Caetano pela confiança e pelos trabalhos desenvolvidos em conjunto. Ao Prof. Dr^o Tiago Alessandro, atual coordenador dessa pós-graduação pela honra de poder participar e contribuir a este programa. Aos demais professores e funcionários dessa instituição que por hora contribuíram diretamente e indiretamente para que esse momento ocorresse.

A Prof. Mercia Eliane, por toda a paciência e dedicação no meu treinamento sobre diagnóstico de malária, meu imenso respeito e admiração. Ao Prof. Dr^o Carlos Alexandre do Centro de Informática da UFPE, pelos ensinamentos relacionados a área de visão computacional e também a sua disponibilidade para sanar algumas dúvidas. Aos Pesquisadores e Professores da Universidade Politécnica da Catalunha, Daniel López, Jordi Serrat e Francesc Zarzuela, pela contribuição, idealização e acompanhamento do projeto.

Por fim à FACEPE pelo apoio financeiro durante toda a pesquisa e também às instituições parceiras: Universidade Politécnica da Catalunha, Universidade Federal Rural de Pernambuco e o Centro de Pesquisas Aggeu Magalhães.

Resumo

A malária é um problema de saúde pública mundial e está relacionada principalmente a áreas remotas. Deste modo, os sistemas de baixo custo para o diagnóstico automático tornam-se uma investigação prioritária em vários grupos de pesquisa. Por outro lado, há novos casos, devido a mudanças climáticas que permitem a sobrevivência do *Anopheles* em áreas antes não habitadas. Os dispositivos móveis têm sido alternativas viáveis em diversos sistemas de saúde e de controle epidemiológico. Os grupos de pesquisa *EpiSchisto Risk Modeling* e *Discrete Modelling and Simulation of Biological Systems* desenvolveram ferramentas de diagnóstico automático para doenças, incluindo a esquistossomose e malária. Neste trabalho, é apresentado um sistema de detecção de parasitos de (malária) a baixo custo com dispositivos móveis, onde o estado da arte envolve duas grandes áreas do conhecimento: na área de computação e uma relacionada à saúde pública. O uso de técnicas de visão computacional e inteligência artificial neste projeto são os fundamentos teóricos do sistema de detecção de malária de baixo custo aqui apresentado. As fases de desenvolvimento do sistema são: captura de imagem - a imagem é capturada usando um sistema de dispositivo móvel experimental acoplado e um microscópio; segmentação - esta etapa consiste em remover a base em torno células do sangue, reduzindo assim o escopo da pesquisa do algoritmo de reconhecimento; treinamento e classificação - utilização de técnicas de inteligência artificial para treinar imagens como: positivos e falsos positivos (com e sem parasitos da malária). Um sistema baseado em Android foi desenvolvido, ele é capaz de detectar células infectadas com *Plasmodium spp* em uma imagem. Este é limitado na identificação do parasito em espécie *P.falciparum* (estágio trofozoíto). A taxa de acerto do sistema em seu melhor desempenho está atualmente em 93%. A principal contribuição do *Malariasystem* é o baixo custo e a possibilidade de ser utilizado por agentes de saúde pública em regiões endêmicas remotas, onde os recursos são escassos. Além disso, os resultados podem ser avaliados in loco, facilitando o tratamento imediato.

Abstract

Malaria is a worldwide public health problem and it is mainly related to remote areas. In this way, low cost systems for automatic diagnosis has become a priority investigation in several research groups. On the other hand, there are new cases due to climate changes which allow the survival of *Anopheles* in areas previously not inhabited. Mobile devices have been viable alternatives in several health systems and epidemiological control. The research groups EpiSchisto Risk Modeling and Discrete Modelling and Simulation of Biological Systems have developed automatic diagnostic tools for diseases including schistosomiasis and malaria. In this work, a detection system for *Plasmodium* parasites (malaria) at low cost with mobile devices is presented, where the state of the art involves two broad areas of knowledge in computing and related to public health. The use of techniques of computer vision and artificial intelligence in this project is the theoretical foundations of the detection system for low-cost malaria presented here. The development phases of the system are: capture image - the image is captured using a coupled experimental mobile device system and a microscope; segmentation - This step consists in removing the background around blood cells, thereby reducing the search scope of the recognition algorithm; training and classification - sorting through artificial intelligence techniques for training with positive and false positive images (with and without malaria parasites). An Android based system was developed, it finds the *Plasmodium* infected parasites in the image. This is limited on the identification of the parasite into *P.falciparum* specie (step trophozoite). The hit rate of the system at its best performance is currently in 93%. The main contribution of *MalariaSystem* is the low cost and the ability to be used by public health officials in remote endemic regions, where resources are scarce. Furthermore the results can be evaluated in loco, facilitating immediate treatment.

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	3
1.3	Contribuições obtidas	3
1.4	Organização do trabalho	4
2	Tópicos sobre malária	5
2.1	Visão sobre malária	5
2.2	Ciclo da doença	6
2.3	Sintomas	9
2.4	Diagnóstico	9
2.4.1	Métodos à coloração de lâminas	10
2.4.2	Características do <i>plasmodium</i>	11
2.5	Tratamento	13
3	Visão computacional	14
3.1	Introdução	14
3.2	Sistema de processamento de imagens	15

3.3	Imagem digital	16
3.4	Sistema visual humano (SVH)	17
3.5	Pré-processamento	18
3.6	Segmentação	19
3.7	Extração de atributos	20
3.7.1	Classificação e reconhecimento	20
4	Fundamentos do <i>Shistosystem</i>	22
4.1	Métodologia de detecção de <i>faces</i>	22
4.1.1	Características	23
4.1.2	Imagem integral	24
4.1.3	Algoritmo de aprendizagem	25
4.1.4	<i>AdaBosting (adaptive boost)</i>	25
4.1.5	Cascata de classificadores	27
5	Modelo proposto - <i>MalariaSystem</i>	30
5.1	Visão geral	30
5.2	Biblioteca <i>OpenCV</i>	31
5.3	Etapa 1: Aquisição da imagem	32
5.4	Etapa 2: Pré-processamento e segmentação	36
5.5	Etapa 3: Treinamento e validação	41
5.5.1	Atividade 1: Criar o conjunto de exemplo positivos e negativos.	41
5.5.2	Atividade 2: Treinar o classificador com os conjuntos de exemplos.	44

5.6	Etapa 4: Desenvolvimento de um sistema de detecção de <i>plasmodium</i> para dispositivos móveis.	45
6	Resultados	49
6.1	Experimentos	50
6.1.1	Experimento 1	51
6.1.2	Experimento 2	53
6.1.3	Experimento 3	55
6.1.4	Experimento 4	57
6.1.5	Experimento 5	58
6.1.6	Curvas ROC <i>Receiver Operating Characteristic</i>	60
7	Conclusões e trabalhos futuros	63
7.1	Contribuições	64
7.2	Limitações	65
7.3	Trabalhos Futuros	65
7.4	Lista de Publicações	67

Lista de Figuras

2.1	Mosquito <i>anopheles darlingi</i>	6
2.2	Ciclo de Vida da Malária Adaptado[1]	8
2.3	Microscópio Bacteriológico Binocular	10
2.4	Estágio Trofozoíta na espécie <i>P.falciparum</i>	12
2.5	Estágio Esquizonte na espécie <i>P.falciparum</i>	12
2.6	Estágio Gametócito na espécie <i>P.falciparum</i>	13
3.1	Etapas de um sistemas de processamento de imagens - Fonte: [2]	15
3.2	Representação matricial. (a)imagem; (b)níveis de cinza correspondentes à região da imagem em destaque	17
3.3	Distribuição de níveis de cinza. (a)imagem; (b)histograma.	18
4.1	Características de <i>haar</i> , adaptado de Viola e Jones [3]	23
4.2	Características de <i>haar</i> , adaptadas de <i>OpenCV</i> [4]	24
4.3	Imagem integral dentro de subjanela - adaptado de Viola e Jones [3]	25
4.4	Cascata de Classificadores - adaptado de Viola e Jones [3]	27
5.1	Fluxo de desenvolvimento do <i>MalariaSystem</i>	30
5.2	Estrutura básica do <i>OpenCV</i> adaptado de [4]	31

5.3	Webcam logitech c270	33
5.4	Camera Sony DSC H1	34
5.5	Samsung galaxy tab 2	34
5.6	Acoplamento experimental utilizando redução hidraulica, onde: (A)Corresponde a peça original, (B)Corte e lixamento e (C)Peça con- cluída.	35
5.7	Acoplamento dos equipamentos ao microscópio	35
5.8	Fluxo Geral	37
5.9	Resultado da remoção de fundo	40
5.10	Software de Marcação de Objetos.	43
5.11	Exemplos de imagens negativas.	43
5.12	Fluxo do aplicativo de detecção por imagem.	46
5.13	Interface do aplicativo de detecção por imagem.	47
5.14	Sistema de detecção no dispositivo móvel.	48
6.1	Curvas ROC onde são representados os seguintes classificadores : (a)Experimento 1 - 94% treinamento e 6% testes com 15 estágios, (b)Experimento 1 - 94% treinamento e 6% testes com 20 estágios, (c)Experimento 5 - 80% treinamento e 20% testes com 20 estágios, (d)Experimento 5 - 94% treinamento e 6% testes com 15 estágios, (e)Experimento 5 - 94% treinamento e 6% testes com 20 estágios.	61

Lista de Tabelas

5.1	Especificações dos equipamentos.	34
6.1	Características padrão a todos os conjuntos de treinamento.	50
6.2	Siglas utilizadas para avaliação.	50
6.3	Dados de entrada do experimento 1	51
6.4	Resultados para o cenário 70% treinamento e 30% validação- experimento 1 .	52
6.5	Resultados para o cenário 80% treinamento e 20% validação- experimento 1 .	52
6.6	Resultados para o cenário 90% treinamento e 10% validação- experimento 1 .	53
6.7	Resultados para o cenário 94% treinamento e 6% validação- experimento 1 .	53
6.8	Dados de entrada do experimento 2	54
6.9	Resultados para o cenário 70% treinamento e 30% validação- experimento 2 .	54
6.10	Resultados para o cenário 80% treinamento e 20% validação- experimento 2 .	55
6.11	Resultados para o cenário 90% treinamento e 10% validação- experimento 2 .	55
6.12	Dados de entrada do experimento 3	56
6.13	Resultados para o cenário 70% treinamento e 30% validação- experimento 3 .	56
6.14	Resultados para o cenário 80% treinamento e 20% validação- experimento 3 .	56
6.15	Resultados para o cenário 90% treinamento e 10% validação- experimento 3 .	57
6.16	Dados de entrada do experimento 4	57

6.17 Resultados - experimento 4	58
6.18 Dados de entrada do experimento 5	58
6.19 Resultados para o cenário 70% treinamento e 30% validação- experimento 5 .	59
6.20 Resultados para o cenário 80% treinamento e 20% validação- experimento 5 .	59
6.21 Resultados para o cenário 94% treinamento e 6% validação- experimento 5 .	60

Lista de Algoritmos

1	Pseudocódigo limiarização simples	19
2	Pseudocódigo do algoritmo <i>AdaBosting</i> , adaptado de Szeliski[5]	26
3	Pseudocódigo da cascata de classificadores, adaptado de Szeliski[5]	29
4	Capturar imagens da camera	41
5	Marcar objetos	42
6	Sistema de detecção de <i>plasmodium Android</i>	48

Lista de símbolos, abreviaturas e Siglas

Símbolos	Descrição
<i>AdaBoost</i>	<i>Adaptative Boost</i>
Android	Sistema operacional para dispositivos móveis
<i>Background</i>	Fundo de uma imagem
<i>Boosting</i>	Algoritmo de aprendizagem supervisionada
C/C++	Linguagem de programação
CPqAM	Centro de Pesquisas Aggeu Magalhães
<i>Faces</i>	facial
<i>Foreground</i>	Objeto de interesse
<i>framework</i>	Arcabouço
Java	Linguagem de programação orientada a objeto
Kernel	Elemento estruturante
NDK	<i>Native Development Kit</i>
OpenCV	<i>Open Computer Vision Library</i>
<i>Pixel</i>	<i>Picture Element</i>
Python	Linguagem de programação
SDK	<i>System Development Kit</i>
USB	<i>Universal Serial Bus</i>
Xml	<i>eXtensible Markup Language</i>
μm	Micrómetro (Unidade de medida)

Capítulo 1

Introdução

Ao longo dos anos, a descoberta de fenômenos físicos e o avanço da tecnologia trouxeram um grande avanço à medicina, proporcionando consigo uma melhora na qualidade de vida para as pessoas. Alguns equipamentos, tais como: Raios-X, ressonância magnética, ultrassom, dentre outros que passam a existir e a auxiliar no diagnóstico e prevenção de enfermidades. A área de processamento de imagens médicas está em grande expansão e trás consigo vários especialistas das mais diversas áreas, tais como: biologia, computação, estatística, engenharia e outras áreas que contribuem e desenvolvem métodos de diagnósticos cada vez mais complexos que auxiliam no diagnóstico de doenças ,por exemplo, câncer e esquistossomose. Nesse sentido algumas áreas da medicina tem se beneficiado, por exemplo, oncologia, cardiologia e ortopedia[2].

A malária é um problema de saúde pública mundial. Em 2010, de acordo com a Organização Mundial de Saúde (OMS), aproximadamente 24 milhões de casos foram diagnosticados com incidência positiva da doença sendo a África o continente mais afetado[6]. A malária é bastante relacionada a áreas remotas. Nesse sentido, um sistema de diagnóstico automático e baixo custo tornou-se uma ferramenta fundamental se quisermos realmente controlá-la. Há casos que surgem em regiões onde não havia prevalência¹ da doença, isto devido às mudanças climáticas que permitem a sobrevivência do mosquito transmissor *Anopheles* em áreas antes não habitadas. Nos próximos anos haverá uma grande demanda por sistemas de diagnóstico e de avaliação *in loco* para estas novas áreas. Seguindo essa linha, o Android[8][9] é uma

¹Em epidemiologia prevalência é a proporção de pessoas em uma população que tem a doença de interesse em um dado ponto ou período de tempo. Fonte: Newman[7]

alternativa viável em vários sistemas de saúde e de controle epidemiológico. Os sistemas computacionais estão cada vez mais robustos e portáteis, portanto trata-se de uma ferramenta auxiliar em ações epidemiológicas e humanitárias[10].

Será visto neste trabalho o desenvolvimento de uma aplicação de detecção de parasitas de malária para dispositivos móveis baseado em etapas de pré-processamento de imagem e reconhecimento de padrões. Em linhas gerais a visão computacional busca auxiliar a resolução de problemas complexos. Para tal, busca simular o sistema visual humano (SVH) e tende a tomar decisões de acordo com as informações contidas nas imagens.

Na área de reconhecimento de padrões em imagens, considera-se o uso de alguns algoritmos clássicos como, Redes Neurais Artificiais, Árvores de Decisão, *Adabosting*, Maquinas de Vetores de Suporte (SVM), K - Vizinhos mais próximos, dentre outros[11][2]. Para este trabalho será apenas utilizado o algoritmo *Adabosting*.

1.1 Motivação

Uma abordagem baseada em *crowd-sourcing* tem sido utilizada para reconhecimento de padrões, que em sua essência busca resolver problemas complexos. Nesta abordagem, este tipo de problema é distribuído a vários participantes que contribuem para a resolução de um dado problema. Existem atualmente várias plataformas de jogos disponíveis para resolver problemas via *crowd-sourcing* em biologia e medicina. Desta forma, permite que pessoas não especializadas na área solucionem problemas específicos. Há na literatura [12] uma abordagem utilizada para reconhecimento de células infectadas por malária. Trata-se de um jogo para reconhecimento de tais células, onde é aplicado a pessoas não especialistas para aprenderem a reconhecer padrões do problema em questão, neste caso células infectadas por malária[12]. Em outra abordagem na literatura [13], é analisada uma imagem contendo parasitos de malária utilizando a técnica de gota espessa (será abordada no Capítulo 2) aliado a um sistema motorizado para mover as lâminas. Anggraini [14] desenvolveu uma aplicação para resolver problemas de segmentação de imagens, conseguindo separar fundo das células sanguíneas. Nesse sentido, A.C Firmo [15] propôs o uso da inteligência artificial e treinamento por enxame de partículas com a finalidade de diagnóstico automático de esquistossomose, sendo realizado junto ao centro de referência em esquistossomose no Centro de Pesquisas Aggeu Magalhães (CPqAM-FIOCRUZ). A partir desse trabalho Oliveira, A.D

[16] realizou um estudo utilizando técnicas de segmentação de imagens afins do diagnóstico automático de malária.

1.2 Objetivos

Esta dissertação busca estudar algoritmos nas áreas de visão computacional e inteligência artificial, com a finalidade do desenvolvimento de um sistema de detecção de malária para dispositivos móveis a baixo custo. Para tanto a presente dissertação tem como objetivos:

- (i) Estudar conceitos de malária, técnicas de visão computacional e inteligência artificial.
- (ii) Desenvolver um modelo utilizando os conceitos e técnicas estudadas em (i), com a finalidade de diagnóstico de malária.
- (iii) Implementar via linguagem de programação o protótipo do modelo desenvolvido em (ii).
- (iv) Implementar e avaliar o uso de dispositivos móveis como ferramenta de apoio a diagnóstico de malária.

1.3 Contribuições obtidas

Nessa dissertação são esperados resultados que possam responder questões, como:

- (i) É possível efetuar uma etapa de segmentação de imagens de modo a extrair informações de objetos dentro das células sanguíneas?
- (ii) De acordo com os fundamentos do *ShistoSystem*[15] é possível prover um bom classificador para diagnóstico para malária?
- (iii) Em dispositivos móveis existe viabilidade para um sistema de baixo custo para diagnóstico de malária?

1.4 Organização do trabalho

Essa dissertação é composta de 7 Capítulos incluindo esta introdução. No Capítulo 2 será feita uma breve revisão acerca dos conceitos básicos da malária, destacando-se o ciclo da doença, técnicas de coloração de lâminas, espécies de plasmodium. No Capítulo 3 serão vistos os conceitos básicos de visão computacional e processamento de imagens, destacando as etapas de um sistema de processamento de imagens. No Capítulo 4 são apresentados os conceitos acerca da metodologia utilizada para classificação e reconhecimento de padrões. Posteriormente no Capítulo 5 será apresentado o modelo desenvolvido para a detecção de malária, destacando-se o fluxo de desenvolvimento, as técnicas utilizadas, e a proposta de dois aplicativos para dispositivos móveis para detecção de plasmodium. No Capítulo 6 são apresentados os resultados relacionados à taxa de acerto do algoritmo implementado. Por fim, o Capítulo 7 apresenta as conclusões e propostas de trabalhos futuros.

Capítulo 2

Tópicos sobre malária

Neste capítulo serão abordados aspectos biológicos, epidemiológicos e sociais da malária. O objetivo geral deste Capítulo é prover um arcabouço conceitual e uma visão sistêmica relacionada à problemática abordada.

2.1 Visão sobre malária

Segundo a Organização Mundial de Saúde (OMS) a malária é considerada um grave problema de saúde pública mundial com aproximadamente 106 países endêmicos¹, tendo a África como continente mais afetado. O seu nome é derivado de *mal aire* ou (*mau ar* em Italiano), acreditava-se que as causas dessa doença estavam relacionadas no ar insalubre de certas regiões pantanosas, porém somente no início do século XX fora descoberto que na verdade existia o papel do inseto ou vetor de transmissão que na verdade é causada por um parasita do gênero *plasmodium*, cinco espécies podem provocar malária nos humanos (*P.falciparum*, *P.vivax*, *P.ovale*, *P.malariae* e *P.knowlesi*). A espécie *P.falciparum* é a mais letal e é predominante na África; *P.vivax* não é tão letal porém é dada como uma espécie mais generalizada ocorrendo em diversas regiões do mundo e atingindo milhões de pessoas. Ao longo dos anos estão sendo medidos esforços para o controle e prevenção da doença, principalmente na região da África Subsariana[18].

¹Em epidemiologia, uma infecção diz-se endêmica, quando ela é mantida nessa população sem necessidade de contaminação proveniente do exterior. Fonte: [17]

No entanto, a malária ainda ocorre em 99 países, causando cerca de 655.000 mortes em 2010 (com uma faixa de incerteza de 537.000-907.000 mortes), principalmente entre crianças menores de 5 anos de idade na África Subsaariana[18][19].

No Capítulo 5 é proposto um modelo para diagnóstico automático de malária, onde utiliza-se duas espécies de *plasmodium*, respectivamente : *P.falciparum* e *P.vivax* ambos na fase trofozoíta. Esta fase será vista nas próximas seções. O Brasil corresponde a praticamente metade de casos de malária na América Latina, tendo como espécie predominante o *P.vivax*. A região amazônica é responsável pela maioria dos casos. Esta contém uma biodiversidade rica e também ambientes propícios à sobrevivência e proliferação do mosquito *anopheles* em especial ao da espécie *anopheles darlingi* vide Figura 2.1 que necessita de locais com águas limpas, quentes, sombreadas e de baixo fluxo[20][6].



Figura 2.1: Mosquito *anopheles darlingi*

O Brasil utiliza a Estratégia Global de Controle Integrado da Malária desde 1993, que basicamente é uma ação conjunta e permanente do governo e da sociedade afins da eliminação ou redução do risco de adoecer ou morrer de malária. Esse modelo foi adotado em recomendação pela Conferência Ministerial de Amsterdã, que aconteceu em 1992[21].

A ferramenta proposta no Capítulo 5 tem a expectativa de auxiliar nessa estratégia global, sendo uma ferramenta de triagem em áreas onde existem poucos recursos (técnicos treinados).

2.2 Ciclo da doença

A malária é transmitida pela picada da fêmea do mosquito *anopheles*, onde estima-se que existam em torno de 70 espécies desse vetor de transmissão capazes de transmitir malária

aos humanos.

O ciclo biológico nas quatro espécies de malária que atingem os humanos é dado em duas etapas. Primeiramente, no vetor de transmissão da doença, o mosquito *anopheles* em uma fase sexuada e posteriormente nos humanos em fase assexuada. O mosquito da espécie *anopheles* tem características alimentares diferenciadas para os machos e as fêmeas. Enquanto os machos alimentam-se de néctar e seiva vegetal, as fêmeas buscam sangue para assim possibilitar o amadurecimento de seus ovos. Isso faz com que somente as fêmeas possam se tornar o vetor de transmissão do *plasmodium*[22].

Nos seres humanos, após a picada do mosquito, inicia-se um ciclo assexuado denominado esquizogônico, onde, através das glândulas salivares, o mosquito libera esporozoítos que são transmitidos ao corpo humano. Logo, estes esporozoítos passam a circular na corrente sanguínea até que penetram nas células do fígado os hepatócitos, a Figura 2.2 representa esse estado em um ciclo denominado esquizogonia hepatócita(A). O tempo de duração de ciclo pode variar de acordo com a espécie de malária que a pessoa esteja infectada, sendo respectivamente: seis dias para o *P.falciparum*, 8 dias para o *P.vivax* e entre 12 a 15 dias para o *P.malariae*. Existe uma peculiaridade nas espécies *P.vivax* e *P.ovale*, ambas podem apresentar um período lento de alguns de seus esporozoítos, formando assim formas latentes e/ou dormentes que são chamados hipnozoítos. Estes são responsáveis por recaídas da doença meses ou até mesmo anos depois da infecção[1][21][6].

No fim desse ciclo ocorre a formação de merozoítos que irão provocar a ruptura dos hepatócitos (etapa (4) da Figura 2.2), fazendo com que estes merozoítos passem a circular na corrente sanguínea. Vale salientar que cada espécie de malária libera uma certa quantidade de merozoítos por hepatócito e podem variar entre 2.000 até 40.000 na espécie *P.falciparum* que é a mais letal da doença[21]. É dado então um novo ciclo denominado equizogonia eritrocitária (visto na Figura 2.2(B)) onde merozoítos passam a circular na corrente sanguínea e invadir as hemácias. Estas possuem como principal função carregar oxigênio aos tecidos do corpo humano.

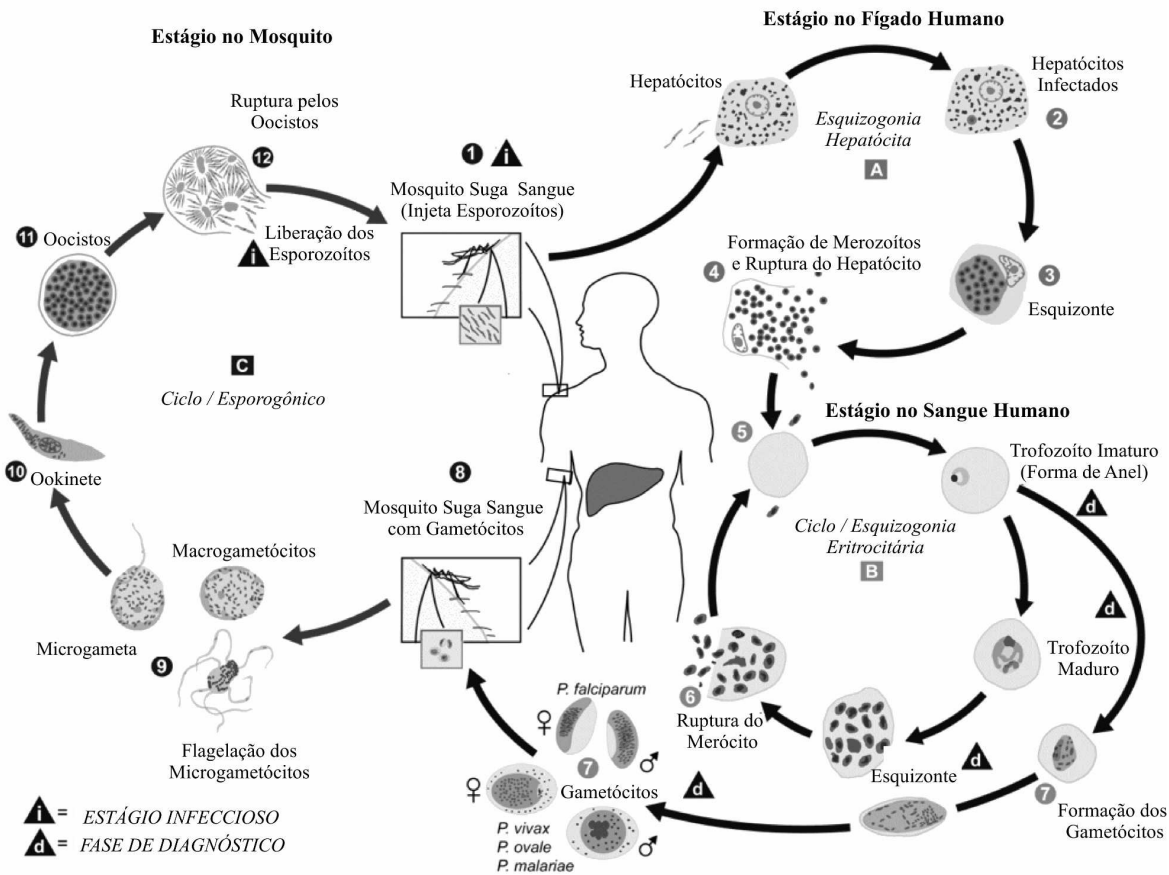


Figura 2.2: Ciclo de Vida da Malária Adaptado[1]

Ao invadir a hemácia, o merozoíto passa a ser um trofozoíto imaturo (forma de anel) onde inicia um ciclo que pode ter um período de 48 a 72 horas. Nesse tempo o parasita se desenvolve dentro da hemácia passando a evoluir para um trofozoíto maduro, mais sem divisão celular. Após esse estágio, inicia-se as divisões celulares passando a ser um esquizonte. O esquizonte origina novos merozoítos que serão lançados na corrente sanguínea após a ruptura do eritrócito, ilustrado na Figura 2.2 (6), iniciando um novo ciclo. O período de 48 e 72 horas pode também ser dito respectivamente como malária terçã e quartã, onde clinicamente os sintomas irão aparecer no período dessa ruptura do eritrócito[1][21].

No ser humano, ocorre também um ciclo denominado sexuado, que é quando após uma série de replicações assexuadas, alguns merozoítos passam a se formar trofozoíto imaturos e posteriormente em gametócitos, onde estes podem ser masculino (microgametócito) ou feminino (macrogametócito). Estes gametócitos têm uma função reprodutiva de perpetuação da espécie dos *plasmodium*[21].

Dando continuidade ao ciclo, esses gametócitos (masculinos e femininos) são ingeridos por novos mosquitos fêmeas durante sua refeição de sangue, dando início a uma etapa sexuada no anopholes. A reprodução sexuada ocorre dentro do estomago do mosquito, os microgametas penetram os macrogametas gerando os zigotos, ilustrado na Figura 2.2(9).

Após uma fusão ele se transforma em um oocineto que migra até a parede do intestino médio do inseto. Depois desse período é gerado um ocisto que dependendo da espécie do mosquito, pode se romper em até 12 dias. Após essa ruptura os esporozoítos são transportados para as glândulas salivares do mosquito, dando continuidade ao ciclo de vida da malária[21].

2.3 Sintomas

A febre é uma característica muito comum da malária, podendo ocorrer no início como terçã ou quartã, com períodos entre 24 e 48 horas de febre geralmente acompanhada de calafrios. Outros sintomas podem vir juntos, tais como: dores de cabeça, fraqueza generalizada, dificuldade respiratória, lesão renal aguda, dores no fígado e no baço[19]. Em algumas situações ela pode ser diagnosticada como gripe, por isso o fator epidemiológico é bastante importante e trata-se de analisar aonde o paciente esteve recentemente e se aquela região é endêmica com malária. Caso a doença não seja diagnosticada rapidamente, seu quadro poderá piorar drasticamente podendo em alguns casos levar até a morte[23].

As manifestações clínicas mais graves com malária aplicam-se em pacientes diagnosticados com a espécie *P.falciparum*. Nesta espécie, a doença pode evoluir ao ponto de provocar a malária cerebral que altera o nível de consciência do paciente, levando-o ao (coma), situação que, se o paciente, sobreviver poderá ter sequelas para o resto da vida[23].

2.4 Diagnóstico

Diagnosticar malária pode ter suas peculiaridades, pois há um elemento importante nesse diagnóstico que é o alto índice de suspeita, fatores como viagens, residência e áreas que essa pessoa se expos devem ser considerados, também não se pode descartar a hipótese de ter ocorrido uma contaminação através de agulhas contaminadas ou transfusões de sangue[19].

O padrão ouro para o diagnóstico de malária é o exame microscópio de sangue, exigindo materiais e reagentes adequados, bem como profissionais treinados capazes de fazer a detecção e diferenciação das espécies de plasmódios[24][21]. O microscópio é um equipamento necessário para este tipo de diagnóstico e à identificação de parasitos do gênero *plasmodium*. Neste caso o microscópio bacteriológico, binocular e com sistema de iluminação incorporado regulado é o mais indicado vide Figura 2.3, no entanto podem ser utilizados outros equipamentos de microscopia, mas sem a garantia da qualidade na visualização de parasitas[24].

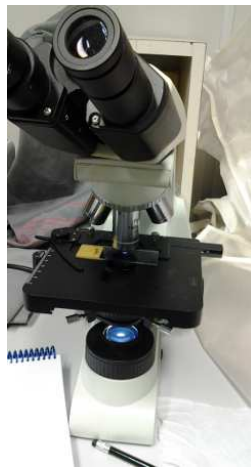


Figura 2.3: Microscópio Bacteriológico Binocular

Na próxima Subseção será visto as técnicas mais conhecidas de coloração de lâminas, estas são essenciais para o diagnóstico de malária através de um microscópio bacteriológico.

2.4.1 Métodos à coloração de lâminas

A técnica de coloração de lâminas é dependente do tipo de diagnóstico laboratorial, onde os mais utilizados são os de gota espessa e esfregaço sanguíneo[24].

A técnica da gota espessa tem a característica de concentrar uma grande quantidade de sangue em uma determinada área, elegendo-a como o método de diagnóstico padrão para malária. Ela é desmembrada e possui um processo de preparação/coloração de lâminas mais rápida. Isso permite uma maior produtividade, ou seja, é possível analisar bem mais amostras nesse método do que comparado a outros métodos de diagnóstico que utilizam microscopia ótica. Como desvantagem, ela requer uma maior experiência por parte do técnico laboratorial com relação à identificação das espécies[24].

Outra técnica bastante difundida de coloração de lâminas é o esfregaço sanguíneo. Quando comparada a outras técnicas como o da gota espessa tem vantagens e desvantagens. Por fixar as hemácias, esta técnica permite um estudo mais detalhado da morfologia, ou seja, permite identificar as características reluzentes a cada espécie do plasmodium, onde não é necessário a desomoglobinização e com isso a perda de parasitas é bem menor comparado à técnica gota espessa. A resistência ao atrito e a durabilidade do material fazem com que as lâminas feitas pelo método do esfregaço sanguíneo tenham uma maior duração na coloração. Como desvantagens ela possui uma menor quantidade de sangue, existindo uma menor possibilidade de encontrar os parasitas, não sendo indicado para o diagnóstico inicial.

No modelo proposto no Capítulo 5 são utilizadas imagens de lâminas contendo parasitas de *plasmodium* utilizando a técnica de esfregaço sanguíneo. A escolha desta técnica se deve a disponibilidade do material para os experimentos, bem como, para em trabalhos futuros ser possível uma etapa de identificação de espécies de *plasmodium*. Na Seção 7.3 é proposto, como trabalhos futuros a identificação de espécies de plasmodium, portanto, as imagens adquiridas neste projeto, bem como o modelo proposto, poderão servir de base para outras pesquisas científicas.

2.4.2 Características do *plasmodium*

O *plasmodium* é um parasita intracelular pequeno com aproximadamente $5\mu m$ (equivalente a $5 * 10^{-6}$ metros) que só pode ser visualizado através do microscópio biológico. Para tal é necessário, à preparação das lâminas de sangue sendo estas coradas pela técnica Giemsa[24]. Como visto na subseção anterior, dois métodos prevalecem quanto à coloração de lâminas: o gota espessa e o esfregaço de sangue [25]. Suas características podem variar de acordo com sua espécie. A espécie *P.facilparum* é a mais letal. Portanto, é importante o diagnóstico precoce para que as chances de sobreviver a doença sejam maiores.

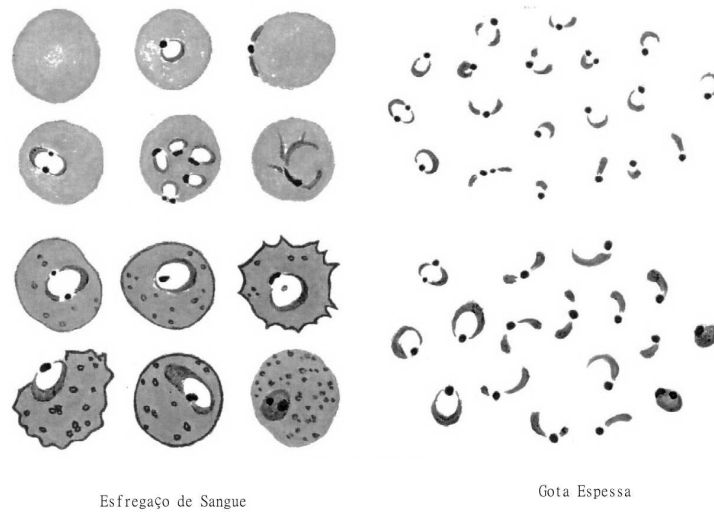


Figura 2.4: Estágio Trofozoíta na espécie *P.falciparum*.

Na Figura 2.4 o *plasmodium* pode ser observado tanto em esfregaço de sangue quanto gota espessa, na fase denominada trofozoíta, uma das etapas do ciclo da doença que já fora visto na Figura 2.2. Os parasitas nessa fase trofozoíta em *P.falciparum* em geral são pequenos em relação às outras espécies de *plasmodium*, numerosos e com formas anelares. Muitas vezes estas podem conter dois pontos tornando uma das características dessa espécie[23].

A fase de Esquizonte em *P.falciparum* é dificilmente vista em lâminas de sangue periférico. Está é uma peculiaridade desta espécie de *plasmodium*. As características em geral são associadas a formas anelares com tamanho pequeno, compacto, e em geral contém entre 12-30 merozoítas agrupados em um pigmento de massa escura e compacta, vide Figura 2.5.

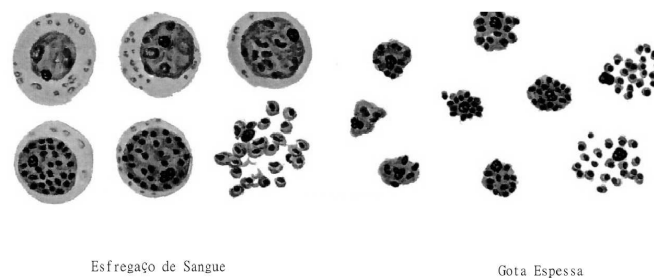


Figura 2.5: Estágio Esquizonte na espécie *P.falciparum*.

Na Figura 2.6 são demonstrados os gametas da espécie *P.falciparum*. Estes apresentam forma de banana ou arredondada (característica dessa espécie), com apenas uma cromatina e bem definida. Seu pigmento assemelha a grãos de arroz, algumas vezes apresenta uma

massa rosada extracelular. Em alguns casos, somente pode-se observar uma cromatina e o pigmento.

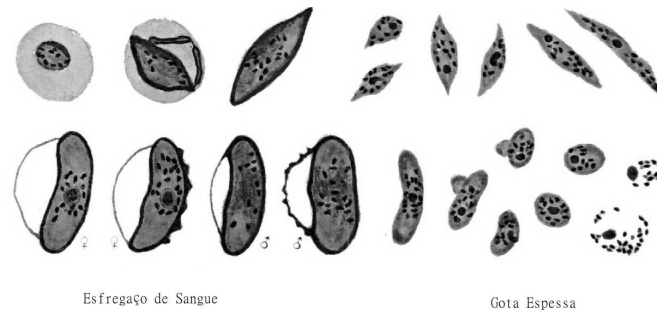


Figura 2.6: Estágio Gametócito na espécie *P.falciparum*.

Conforme dito anteriormente para o modelo proposto no Capítulo 5 é utilizado o estágio trofozoíta, devido a disponibilidade de lâminas de *plasmodium* contendo esta etapa.

2.5 Tratamento

O tratamento é realizado por meio de medicamentos antimaláricos como a Cloroquina ou Primaquina. As dosagens dependem de fatores que vão desde o peso do paciente até o nível de infecção e a espécie do plasmodium. A espécie *P.falciparum* é a mais letal e merece um cuidado especial quanto ao acompanhamento deste paciente[23]. O guia de referência em tratamento de malária[21] indica que antes de informar qual medicamento e a dosagem, são feitos os seguintes questionamentos :

- Espécie de plasmódio infectante, pela especificidade dos esquemas terapêuticos a serem utilizados;
- Idade do paciente, pela maior toxicidade para crianças e idosos;
- História de exposição anterior à infecção, uma vez que indivíduos primo infectados tendem a apresentar formas mais graves da doença;
- Condições associadas, tais como gravidez e outros problemas de saúde;
- Gravidade da doença, pela necessidade de hospitalização e de tratamento com esquemas especiais de antimaláricos.

Capítulo 3

Visão computacional

Neste Capítulo serão abordados os principais conceitos utilizados neste trabalho sobre processamento de imagens e visão computacional, tendo como objetivo conhecer as diversas ferramentas e técnicas utilizadas neste trabalho, proposto nesta dissertação na implementação do sistema de detecção de parasitos *plamosdium* em amostras de sangue.

3.1 Introdução

A crescente evolução dos sistemas computacionais e da tecnologia da informação de uma forma geral propuseram benefícios à toda humanidade. Passaram a existir diversas ferramentas de processamento de dados. Acompanhando esse crescimento surgiu o processamento de imagens, que nada mais é do que um conjunto de algoritmos que, através de formulas matemáticas conseguem modificar e/ou alterar imagens sucessivamente[26][2].

Segundo Gonzales e Woods [26] “O interesse em métodos de processamento de imagens digitais decorre de duas áreas principais de aplicação: melhoria de informação visual para a interpretação humana e o processamento de dados de cenas para percepção automática através de máquinas”.

Segundo Predrini [2], o ser humano em si possui uma grande capacidade de captar, processar e interpretar grandes volumes de dados. O estudo dos estímulos visuais dos seres humanos possui uma grande importância ao desenvolvimento de novas técnicas ou até mesmo ao aprimoramento das já existentes no que diz respeito à análise automática ou

semiautomática de informações extraídas de uma imagem ou vídeo com o auxílio de um equipamento computacional.

3.2 Sistema de processamento de imagens

Um sistema de processamento digital de imagens é constituído de várias etapas com o objetivo de gerar um resultado a partir de um dado como imagem de entrada. A Figura 3.1 ilustra as etapas conforme [2].

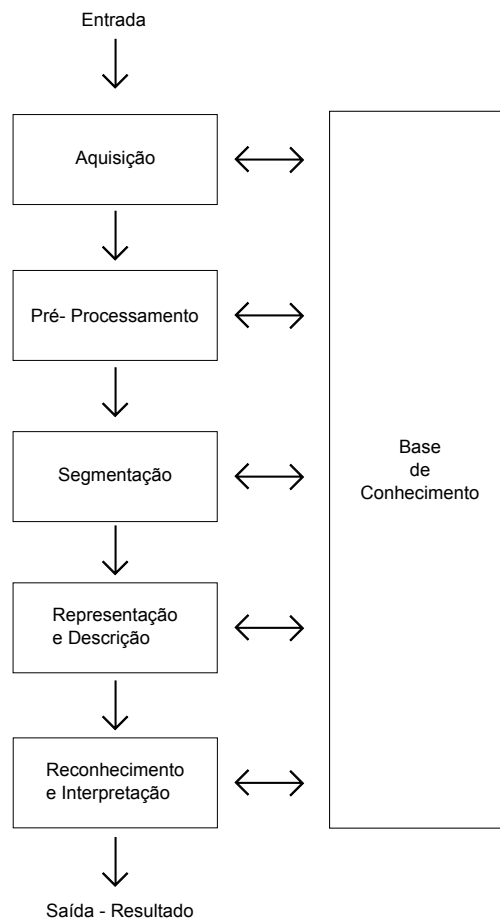


Figura 3.1: Etapas de um sistemas de processamento de imagens - Fonte: [2]

A etapa de aquisição captura a imagem por meio de um dispositivo ou sensor, por exemplo: câmeras digitais, tomógrafos médicos, dentre outros. A escolha do dispositivo de aquisição é um processo importante para todo o sistema de processamento de imagens, uma vez que as outras etapas dependem desta etapa para serem executadas. Fatores como iluminação e qualidade do sensor podem influenciar para as etapas seguintes. Mesmo com

todos os cuidados na etapa anterior da aquisição da imagem, falhas podem ocorrer, que vão desde problemas com iluminação ou até mesmos ruídos em imagens. Para armazenar esses efeitos é dada a etapa de pré-processamento, que tem como objetivo realizar correções nas imagens, que podem ser, correção de contraste, suavização na imagem, dentre outras.

A segmentação é uma etapa considerável no que diz respeito à extração de características e/ou informações em imagens, principalmente no âmbito médico que por muitas vezes são necessários para separar regiões da imagem e posteriormente classificá-las, a fim de encontrar anomalias ou outros elementos de interesse. Em algumas aplicações, a segmentação e o processamento de imagem pode ser aplicado para pesquisa de vírus e até mesmo identificar HIV por exemplo; Outra possibilidade seria no âmbito da neurocirurgia e outras aplicações médicas[27].

Em linhas gerais, a última etapa de um sistema de processamento digital de imagens é classificar e reconhecer o objeto de interesse a partir dos atributos extraídos na etapa de extração de características. Após esta etapa é dado o resultado final do sistema, com as imagens rotuladas e classificadas.

Em todas as etapas pode haver um auxílio externo e não pertencente ao fluxo, denominado base de conhecimento, que é um guia para a comunicação entre os módulos de processamento a fim de executar uma determinada tarefa, que é simplesmente o conhecimento sobre o domínio geral do problema. No Capítulo 5 é proposto um fluxo com base na Figura 3.1 para a detecção de parasitos do tipo *plasmodium*.

3.3 Imagem digital

Uma imagem digital bidimensional monocromática é uma função $f(x, y)$ discreta que recebe tanto coordenadas espaciais quanto em brilho. Ela pode ser entendida como uma matriz cujo índice de linhas e colunas indica um ponto ou posição naquela imagem e o correspondente valor desse elemento dessa matriz condiz ao nível de cinza daquele determinado ponto. A estes pequenos pontos é dado o nome de *pixel* abreviação de *picture elements*, ou seja, elementos da Figura[26].

Em imagens digitais a função $f(x, y)$ deve ser convertida para a forma discreta. Desta forma, é dado um processo denominado *digitalização*, que envolve *amostragem* e *quantização*.

Segundo Pedrini [2] a amostragem consiste em discretizar o domínio de definição da imagem em x e y , com o objetivo de gerar uma matriz de $M \times N$ amostras. Já a quantização consiste em escolher o número de inteiros L de níveis de cinzas, No caso da imagem monocromática a cada ponto da imagem. A Figura 3.2 ilustra esse processo.

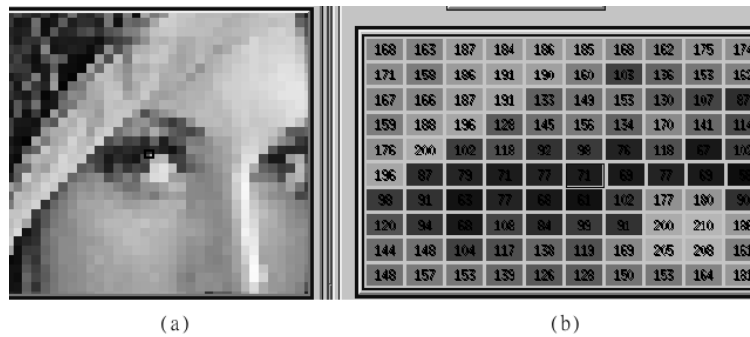


Figura 3.2: Representação matricular. (a)imagem; (b)níveis de cinza correspondentes à região da imagem em destaque

As imagens adquiridas no Capítulo 5 são digitais assumindo uma função $f(x, y)$, onde quando convertida para a escala de cinza é gerada uma matriz com os valores de intensidades dos pixels. Esta operação é dada com o uso de algoritmos da biblioteca de visão computacional OpenCV[4].

3.4 Sistema visual humano (SVH)

O ser humano tem capacidades sensoriais para adaptação e percepção de ambientes. Dentre os sentidos a visão é um dos mais importantes para essa habilidade, pois ela envolve algumas funções essenciais, tais como: detecção, localização, reconhecimento e por fim interpretação de objetos. Com essas funções, a visão computacional busca entender os fenômenos da compreensão humana para auxiliar no desenvolvimento de sistemas capazes de adquirir, analisar e interpretar informações e estímulos visuais, afins de aperfeiçoar ou até mesmo criar novas técnicas para realizar essas funções através de máquinas[2].

Uma das partes mais importantes do olho humano é a retina, ela é composta de células fotossensíveis que captam e processam informação visual, são elas: bastonetes, responsáveis para a adaptação à luz noturna e os cones, que captam informação luminosa vinda da luz do dia, das cores e do contraste[27].

3.5 Pré-processamento

Técnicas de pré-processamento em imagens buscam melhorar a aparência ou até mesmo simplificam uma imagem, ou seja, corrigir e eliminar ruídos que geralmente são advindos dos equipamentos de aquisição, bem como degradações nas imagens ocasionadas por problemas de iluminação. São dados dois domínios para realces em imagens: o espacial, que se refere ao próprio plano da imagem trabalhando em cima de *pixels*, e o domínio de frequência, que são modificações em imagens com aplicações da transformada de *Fourier*[2].

Há na literatura[2][27][26][5] uma representação para obter informações sobre a distribuição de pixels no domínio espacial, que é o calculo do *histograma*, ilustrado na Figura 3.3.

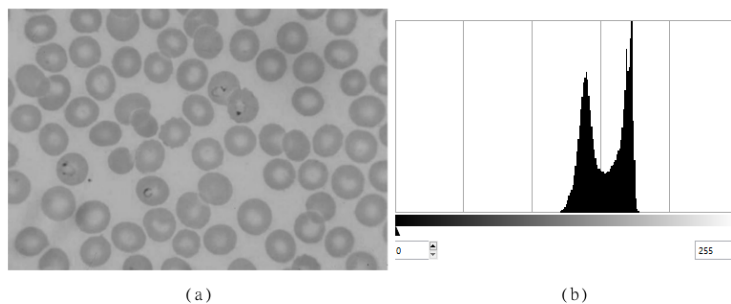


Figura 3.3: Distribuição de níveis de cinza. (a)imagem; (b)histograma.

Existem operações para distribuições dos pixels como, por exemplo, a equalização de histograma, que consiste em mapear os valores das intensidades dos pixels de uma região com pouco contraste até uma com muito contraste, de forma a distribuir uniformemente os pixels no decorrer de uma dada imagem[2].

Segundo Pedrini [2] uma imagem colorida é uma imagem multibanda ou multiespectral, onde cada cor nos pixels $f(x, y)$ é definida por três grandezas: *luminância*, *matiz* e *saturação*. A *luminância* é associada ao brilho da luz, o *matiz* o comprimento da onda dominante e a *saturação* com o grau de pureza luz branca do matiz.

3.6 Segmentação

Interpretar dados digitais é uma tarefa complexa e custosa computacionalmente [27], segmentar ou particionar uma imagem digital em partes menores é uma tarefa essencial para auxiliar o processo de interpretação de imagens. As imagens em geral podem conter características distintas o que por muitas vezes dificultam esse processo. Imagens com diferentes tamanhos, diferenças de iluminação, distorção de formas, entre outras, são consideradas ruídos em imagens, ou seja, são dados contidos nas imagens que dificultam, de certa forma, a identificação de características contidas nessas imagens, por exemplo, a captura de imagens de placas de automóveis é realizada em um ambiente não controlado, ou seja, existe variação de iluminação de acordo com o horário e as condições climáticas[2].

Existe uma grande variedade de técnicas de segmentação de imagens, uma das mais simples e bastante utilizadas é a de limiarização ou binarização, que consiste na classificação de pixels de uma dada imagem de acordo com um ou vários limiares. Essa técnica é bastante utilizada para a separação entre background (fundo) e objeto de interesse. Elas são classificadas em limiarização global, onde um único valor de corte é definido para toda a imagem. Algumas vantagens são: a velocidade, porém podem perder em qualidade, uma vez que uma imagem digital pode conter ruídos que interfiram no processo. O outro tipo de limiarização é a local, esta consiste em ter diferentes pontos de corte para diferentes regiões da imagem, uma desvantagem é o tempo de processamento alto. Um algoritmo simples de limiarização baseado no corte de um limiar através dos tons de cinza de uma imagem é ilustrado no pseudocódigo 1 [26][2].

Algoritmo 1 Pseudocódigo limiarização simples

```
1: cinza = matriz  $MXN$  de uma dada imagem em tons de cinza
2: para  $i \leftarrow$  até todos os pixels de cinza faça
3:   se  $cinza[i] \leq 127$  então
4:      $cinza =$  preto
5:   senão
6:      $cinza =$  branco
7:   fim se
8: fim para
```

Em uma imagem em tons de cinza, assumi-se 256 níveis de intensidade, portanto de acordo com o pseudocódigo 1, quando um pixel tem o valor igual ou inferior a (127) o valor desse pixel será (0) preto, caso contrário os pixels serão classificados como branco (255).

Desta forma, temos uma imagem binária ou limiarizada. Outras técnicas de segmentação são vistas na literatura, tais como: segmentação de regiões, divisor de águas[2]. As técnicas de segmentação utilizadas para o modelo proposto no Capítulo 5 serão apresentadas na Seção 5.4.

3.7 Extração de atributos

Após uma imagem pré-processada e segmentada, torna-se possível extrair atributos, atributos de uma dada região e ou objeto de interesse. Estas propriedades podem ser utilizadas para a discriminação entre classes, em geral são atributos numéricos que formam um vetor com características. Outras propriedades podem ser extraídas de imagens, tais como: o número de objetos, propriedades geométricas, dimensões, análise de textura, objetos destacados[2].

3.7.1 Classificação e reconhecimento

Com as definições de atributos e características é necessário distinguir objetos na imagem, de tal forma a agrupar de acordo com semelhança para cada região de *pixels*. Uma aplicação prática pode ser dita como um reconhecimento automático de objetos. Estas podem levar em considerações atributos extraídos na etapa anterior de extração de atributos, por exemplo, dada uma imagem com três formas geométricas distintas o objetivo é agrupar os objetos semelhantes; portanto, temos três classes distintas com um conjunto $Y = w1, w2, w3$. Segundo Pedrini [2] “os algoritmos que visam estabelecer o mapeamento entre as propriedades de amostras e o conjunto de rótulos são denominados como algoritmos de classificação ou classificadores”.

A etapa de classificação pode ser denominada como *reconhecimento de padrões*, sendo parte da área de aprendizado de máquina onde o objetivo é distinguir objetos (padrões) em categorias (classes). Essa tarefa como um todo não é simples, e pode envolver várias etapas. Os algoritmos de classificação podem ser divididos em *supervisionados* e *não-supervisionados*[28].

Em uma classificação supervisionada assumi-se o fato de existir um conhecimento prévio

da classe a ser reconhecida. Para tal, é necessária uma etapa denominada treinamento antes da execução do algoritmo de classificação de objetos, aonde é dado um conjunto de amostras identificadas que devem conter suas propriedades e a classe a qual ela pertence. Essas amostras são denominadas conjunto de treinamento[2].

Por outro lado a classificação não supervisionada é utilizada quando não se dispõe de informações e parâmetros coletados previamente; porém, durante a classificação, os objetos com características semelhantes recebem um rótulo contendo a classe do objeto. Este aprendizado também é conhecido como agrupamento ou *clusterização*, onde, na maioria das vezes, a quantidade de classes, bem como suas respectivas condições iniciais, são dadas como entrada para um algoritmo de treinamento, que busca por grupos representativos em um dado conjunto de dados[2].

Há várias técnicas de aprendizagem de máquinas, dentre elas: K-vizinhos mais próximos, redes neurais, árvore de decisão, suporte a vetores de decisão[2][27]. O algoritmo utilizado neste trabalho é do tipo supervisionado e será visto no Capítulo 4.

Capítulo 4

Fundamentos do *Shistosystem*

Um sistema de diagnóstico automático de esquistossomose foi proposto por A.C Firmo [15], e consiste em identificar e realizar a contagem de ovos de *Schistosoma Mansoni* a partir da proposta de um dispositivo de baixo custo. Neste caso foi realizado com uma câmera acoplada a um microscópio bacteriológico. Os fundamentos do trabalho [15] baseiam-se na técnica de detecção de faces proposta por Viola e Jones [3] aliados a uma otimização por exame de partículas.

Neste Capítulo serão vistos os fundamentos do *Shistosystem* com base na metodologia de detecção de faces de Viola e Jones [3] e que foi utilizado neste trabalho como fundamentação para o *MalariaSystem*. Entretanto foram realizadas adaptações que serão apresentadas no Capítulo 5.

4.1 Metodologia de detecção de *faces*

O método de reconhecimento facial proposto por Viola e Jones [3] é conhecido mundialmente como uma heurística para a proposta de detecção de rostos e ou *faces* em imagens. No entanto, outros estudos como AC. Firmo [15] e de Oliveira A.D [16] mostraram a aplicação desta técnica para a detecção da doença em lâminas de laboratório.

A proposta de Viola e Jones [3] tem como principal objetivo uma forma robusta e rápida de detecção de rostos e *faces* com altas taxas de detecção. Uma das considerações e contribuições desse método é uma nova representação da imagem, denominada *imagem integral*,

seguido por um classificador simples e eficiente que utiliza um algoritmo de aprendizado *ada-boost*. Outra importante contribuição é a cascata de classificadores. A metodologia tem um custo computacional $O(1)$, ou seja, tempo constante, o que permite seu uso em aplicações de tempo real. Essas contribuições serão abordadas nas próximas seções deste capítulo.

4.1.1 Características

Historicamente trabalhar diretamente nos pixels sempre foi muito custoso em nível de complexidade computacional[26]. Viola e Jones [3] em seus objetivos buscavam construir um sistema que pudesse ser executado em tempo constante, diante disto sugeriu uma adaptação nas funções básicas de *haar* descritas por Papageorgiou [29] e passou a utilizar em seu trabalho denominando características de *haar* ou *haar-like*. As características de *haar* foram aliadas ao uso da *imagem integral* que será vista na próxima seção.

Basicamente, as características de *haar* são retângulos compostos por duas regiões uma clara e outra escura. O valor da característica é a diferença entre a soma das intensidades dos pixels da região clara e a soma das intensidades dos pixels da região escura. A seguinte fórmula é dada na Equação (4.1):

$$f(w) = \sum \mathbf{P}_{preto} - \sum \mathbf{P}_{branco} \quad (4.1)$$

Onde:

$f(w)$ = valor do *feature* na janela w ;

$\sum \mathbf{P}_{preto}$ = Somatório de pixels na região preta;

$\sum \mathbf{P}_{branco}$ = Somatório de pixels na região branca.

Em seu trabalho Viola e Jones [3] descreve as quatro características básicas de *haar* para utilizar em detecção de rostos e ou faces, conforme Figura (4.1).

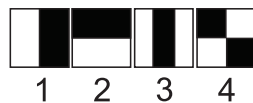


Figura 4.1: Características de *haar*, adaptado de Viola e Jones [3]

Cada tipo de recurso das características de *haar* pode indicar a presença ou ausência de algumas características de uma dada imagem, como por exemplo: bordas, região de fronteira de iluminação, textura, dentre outras. O detector em sua forma padrão funciona na resolução de 24 X 24 pixels gerando até 160.000 retângulos(características) por subjanela [29][3].

Em versões mais recentes, um novo conjunto de características de retângulos é proposto, ilustrado na Figura 4.2. Além de novas características acrescentadas, as quatro originais vistos na Figura 4.1, estão inclusas e oferecem versões com rotação. Para mais detalhes dos cálculos envolvendo as rotações da características de *haar*, consultar [30][4].

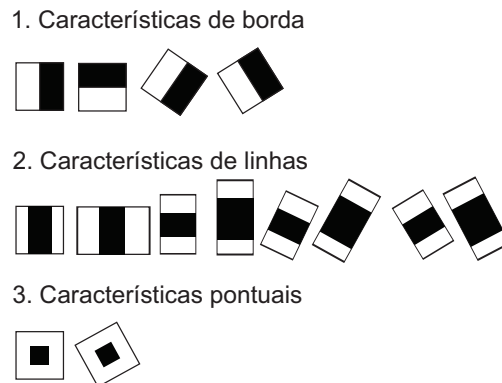


Figura 4.2: Características de *haar*, adaptadas de *OpenCV* [4]

4.1.2 Imagem integral

Imagem integral é uma representação que pode ser utilizada onde se tem uma função de pixels para números reais $f(x, y)$, como por exemplo, a *intensidade de pixels* e queremos calcular a soma dessa função ao longo de uma região retangular da imagem. A fim de computar rapidamente os retângulos das características de *haar*, Viola e Jones [3] propôs o uso da imagem integral que é dado pela equação 4.2.

$$ii(x, y) = \sum_{x^l \leq x, y^l \leq y} i(x^l, y^l), \quad (4.2)$$

Onde, $ii(x, y)$ é a imagem integral e $i(x, y)$ é a imagem original. Para cada ponto (x, y) é atribuído o valor da soma de todos os pixels superiores e para a esquerda ,tais que $x^l \leq x, y^l \leq y$.

A imagem integral pode ser calculada apenas com uma única passagem sobre a imagem, usando o valor da área somada em (x, y) , conforme Equação (4.3):

$$I(x, y) = f(x, y) + I(x - 1, y) + I(x, y - 1) - I(x - 1, y - 1) \quad (4.3)$$

Uma vez calculada a imagem integral, avaliar qualquer área do retângulo pode ser realizada em tempo constante $O(1)$ com apenas quatro referências à matriz, conforme Figura (4.3).

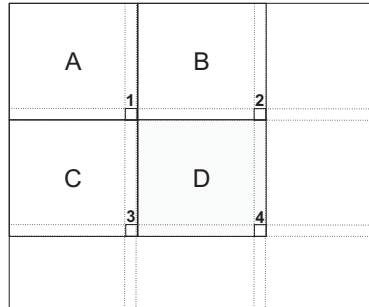


Figura 4.3: Imagem integral dentro de subjanela - adaptado de Viola e Jones [3]

A partir da imagem integral, a soma da região D pode ser calculada usando apenas os valores da imagem integral nos vértices aqui exemplificados por 1, 2, 3 e 4, conforme calculo visto na Equação (4.4).

$$D = II(4) - II(2) - II(3) + II(1) \quad (4.4)$$

4.1.3 Algoritmo de aprendizagem

A proposta de Viola e Jones [3] utiliza a ideia de *classificadores fracos* ($h(x, f, p, \theta)$) que em definição, nada mais é do que uma estrutura simples que contém uma subjanela (x), um vetor de características (f), um limiar (θ), e uma paridade (p).

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{se } pf(x) < p\theta \\ 0 & \text{caso contrario} \end{cases} \quad (4.5)$$

O algoritmo de aprendizagem nessa etapa tem o objetivo de selecionar uma única característica que melhor separa os exemplos positivos e negativos. Para tal é dado um algoritmo de impulsionar baseado em *boosting*[31].

4.1.4 *AdaBoosting (adaptive boost)*

Adaboosting é um meta-algoritmo de aprendizado de máquina proposto por Freund e Schapire[31] onde a saída de outros algoritmos de aprendizagem chamado de (alunos fracos)

é combinado com uma soma ponderada que representa o resultado final do classificador potencial. Tem como principal característica a distribuição de pesos nos conjuntos de exemplos e a modificação da distribuição com o decorrer das iterações do algoritmo.

Algoritmo 2 Pseudocódigo do algoritmo *AdaBoosting*, adaptado de Szeliski[5]

- 1: Inserir os exemplos positivos e negativos e seus rotulos (x_i, y_i) , onde $y_i = 1$ = exemplos positivos e $y_i = -1$ = exemplos negativos.
- 2: Inicializa os pesos para $w_i, 1 \leftarrow \frac{1}{N}$, onde N é o numero de treinamento dos exemplos.
- 3: **para** $t = 1 \leftarrow$ **até** T **faça**
- 4: Normalizar os pesos de modo que eles somem 1 (Dividi-los por sua soma)
- 5: Selecionar o melhor classificador $h_j(x; f_j, \theta_j, s_j)$ afim de encontrar o que minimiza o erro de classificação ponderada

$$e_j = \sum_{i=0}^{N-1} w_i \cdot t \cdot e^{-y_i \cdot h(x_i)} \quad (4.6)$$

$$e_{i,j} = 1 - \delta(y_i, h_j(x_i, f_j, \theta_j, s_j)). \quad (4.7)$$

- 6: Calcule a taxa de erro β_j e o peso do classificador α_j

$$\beta_j = \frac{e_j}{1 - e_j} \quad e \quad \alpha_j = \frac{1}{2} \log(-\beta_j) \quad (4.8)$$

- 7: Atualizar os pesos de acordo com os erros de classificação $e_{i,j}$

$$w_{i,j+1} \leftarrow w_{i,j} \beta_j^{1-e_{i,j}} \quad (4.9)$$

- 8: Definir o classificador final

$$f(x) = \sin \left[\sum_{j=0}^{m-1} \alpha_j h_j(x) \right] \quad (4.10)$$

- 9: **fim para**
-

Conforme pseudocódigo 2 o *adaboost* recebe inicialmente como entrada do algoritmo um conjunto de treinamento $(x_1, y_1) \dots (x_n, y_n)$, onde cada x_i pertence a algum espaço do domínio ou instancia de X , e cada rótulo y_i é definido por Y . O *AdaBoost* chama um algoritmo de aprendizagem chamado aluno fraco determinado repetidamente em uma série de rodadas $t = 1, \dots, T$. O algoritmo sempre busca manter uma distribuição ou conjunto de pesos sobre o conjunto de treinamento. Inicialmente, todos os pesos são definidos igualmente, mas em cada rodada, os pesos dos exemplos incorretamente classificados são aumentados para que o aluno fraco seja forçado a concentrar-se nos exemplos do conjunto de treinamento. A cada

iteração, os exemplos são ordenados pelo valor do limiar, a escolha de um limiar ótimo não é subjetiva[15]. O limiar ótimo para cada característica pode ser encontrado através de uma única passagem pela lista ordenada.

4.1.5 Cascata de classificadores

Nas tarefas de classificação em geral há um custo elevado para avaliação de todo o conjunto de treinamento, portanto a redução da quantidade de classificadores tende a melhorar a velocidade de um sistema deste tipo. No entanto, a diminuição desses classificadores reduz a taxa de acerto. Desta forma, Viola e Jones [3] propôs uma solução afim de obter um cenário diferenciado em que a escolha de bons classificadores em um grande conjunto e o alinhamento destes tornou possível a criação de árvores de decisão denominadas por ele de cascata de classificadores.

Em linhas gerais uma cascata inicialmente busca classificadores mais simples (genéricos) que tem o objetivo de rejeitar o maior número de sub-janelas negativas possíveis. Desta forma, é possível realizar uma filtragem inicial. Ao decorrer das etapas da cascata são dados classificadores mais complexos, a classificação de certa forma tende a se tornar mais fina com o aumento da precisão do algoritmo; ou seja, em um estágio inicial da cascata são avaliadas poucas características. A medida que o processo avança o número de características tende a crescer e melhorar a precisão dos classificadores, porém não é uma regra. Na Figura 4.4 é ilustrada uma cascata de classificadores.

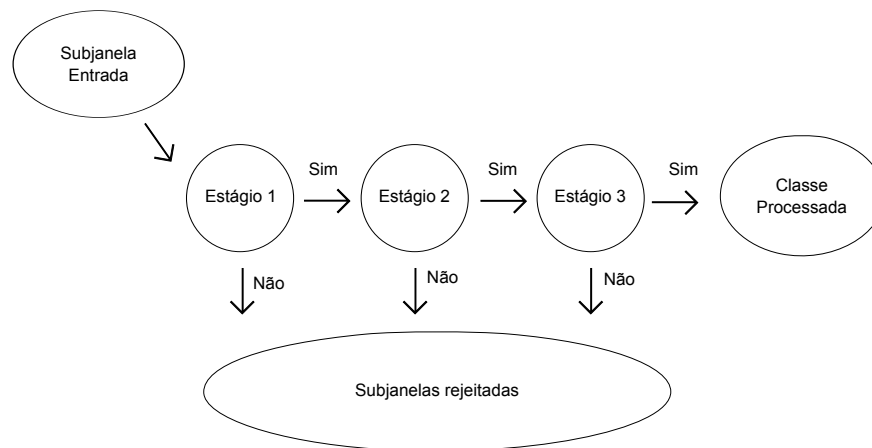


Figura 4.4: Cascata de Classificadores - adaptado de Viola e Jones [3]

Conforme a Figura 4.4, o processo de detecção de uma cascata de classificadores tem

estrutura de uma árvore de decisão degenerada. Portanto, um resultado positivo no primeiro classificador ativa a avaliação do segundo classificador que, a medida que os estágios da cascata crescem, é aumentado o número de características de *haar* e conseqüentemente uma melhor precisão na taxa de detecção. Por outro lado, um resultado negativo em qualquer ponto da cascata conduz à rejeição imediata de uma sub-janela analisada.

Em seus estágios iniciais, com o classificador mais genérico, a cascata busca rejeitar o maior número de sub-janelas negativas, uma vez que no caso de detecção de faces, o número de sub-janelas negativas é maior do que a de positivas. Com isso, conforme dito anteriormente é feita uma filtragem a medida que os classificadores avançam em cascata[3][15][5]. Segundo Viola e Jones [3] o uso da cascata de classificadores tende a aumentar a velocidade na etapa de classificação. O número de estágios da cascata e o tamanho de cada um deles deve ser suficiente para alcançar um desempenho aceitável. A taxa de falso positivos é um desses parâmetros, sendo ela o produto de todas as taxas de falso positivos encontrados nos nós de sua árvore, conforme Equação (4.11).

$$F = \prod_{i=1}^k f_i \quad (4.11)$$

onde, F é a taxa de falsos positivos do classificador; k o número de classificadores e f_i a taxa de falsos positivos do i -ésimo classificador. A taxa mínima de detecção aceitável é dado pelo seguinte produtório:

$$D = \prod_{i=1}^k d_i \quad (4.12)$$

onde, D é a taxa de detecção do classificador em cascata; k o número de classificadores e d_i a taxa de detecção do i -ésimo classificador. No Algoritmo 3 é ilustrado o pseudocódigo de uma cascata de classificadores.

Algoritmo 3 Pseudocódigo da cascata de classificadores, adaptado de Szeliski[5]

-
- 1: Selecionar os valores para f , taxa máxima de falsos positivos por fase, e d a taxa mínima de detecções por fase.
 - 2: Selecionar a taxa de falso positivos global
 - 3: P = Conjunto de exemplos positivos
 - 4: N = Conjunto de exemplos negativos
 - 5: $F_0 = 1, 0; D_0 = 1, 0; i = 0$.
 - 6: **enquanto** $F_i > F_{global}$ **faça**
 - 7: $i = i + 1$
 - 8: $n_i = 0; F_i = F_{i-1}$
 - 9: **enquanto** $F_i > f(F_{i-1})$ **faça**
 - 10: $n_i = n_i + 1$
 - 11: Usar P e N para treinar o classificador com n_i características, usando o *Adaboost*.
 - 12: Avaliar o classificador atual com um conjunto de validação para determinar F_i e D_i .
 - 13: Diminuir o limiar do i -ésimo classificador até que a cascata atual tenha uma taxa de detecção pelo menos $d(D_{i-1})$ (afetando também F_i).
 - 14: **fim enquanto**
 - 15: $N = 0$
 - 16: **se** $F_i > F_{global}$ **então**
 - 17: Avaliar detector atual em cascata num conjunto de imagens que não contêm *faces* e inserir as que forem mal classificadas dentro do conjunto N .
 - 18: **fim se**
 - 19: **fim enquanto**
-

A taxa máxima aceitável para f_i e a taxa mínima aceitável para d_i , em cada estágio da cascata, é treinado pelo *adaboost* com o número de características adequadas para uma taxa de detecção designada. Por exemplo, para uma taxa de detecção de 98% é dado $0.999^{20} \approx 0,98$. As taxas designadas para o classificador são testadas em um conjunto de validação, onde, se a taxa de falsos positivos não for atingida, uma nova característica será acrescentada a cascata até que as taxas mínimas sejam atingidas [3].

Neste capítulo, foi apresentada a metodologia de detecção de faces proposta por Viola e Jones[3]. No próximo Capítulo será apresentado um fluxo contendo as etapas de desenvolvimento do modelo proposto *Malariasystem*.

Capítulo 5

Modelo proposto - *MalariaSystem*

Neste capítulo, será apresentando o modelo proposto para este trabalho. Este modelo é dado em várias etapas de visão computacional, a fim de identificar parasitos do gênero *plasmodium* em imagens.

5.1 Visão geral

A solução proposta nesse projeto é denominada *MalariaSystem* e obedece um fluxo de desenvolvimento de acordo com técnicas de visão computacional abordadas, e com os fundamentos do *ShistoSystem* proposto por AC.Firmo [15].

O projeto é dividido em quatro etapas principais, a primeira consiste na aquisição das imagens, onde é proposto o acoplamento experimental de uma câmera a um microscópio bacteriológico; a segunda a fase é o pré-processamento e a segmentação dessas imagens capturadas, a terceira fase trata-se de um sistema de treinamento e validação e a quarta etapa é o desenvolvimento de um sistema de detecção de *plasmodium* para dispositivos móveis , fluxo vide Figura 5.1.

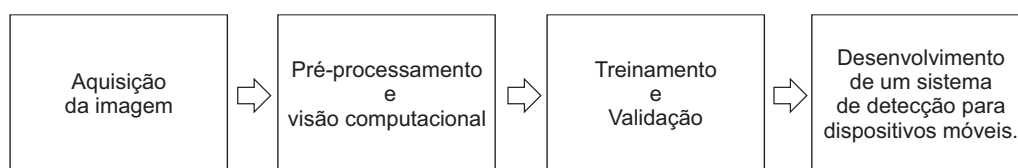


Figura 5.1: Fluxo de desenvolvimento do *MalariaSystem*

5.2 Biblioteca *OpenCV*

O *OpenCV* (*Open Computer Vision Library*) é uma biblioteca de visão computacional multiplataforma de uso livre, ela é desenvolvida em C e C++ fazendo com que suas aplicações obtenham um bom desempenho e que possam ser utilizadas para sistemas de tempo real, ela possui interfaces para Python, C, C++, Ruby, Matlab, dentre outras linguagens. Essa biblioteca é otimizada para os processadores de arquitetura Intel[4].

Essa biblioteca foi projetada para ter um bom desempenho em aplicações de tempo real, um dos seus objetivos é fornecer uma infraestrutura de visão computacional simples que auxilia os desenvolvedores a criar aplicações de visão computacional bastante sofisticadas. Ela contém mais de 500 funções que podem ser úteis para diversas aplicações tais como: imagens médicas, segurança, interface com usuário, robótica, dentre outras[4][32].

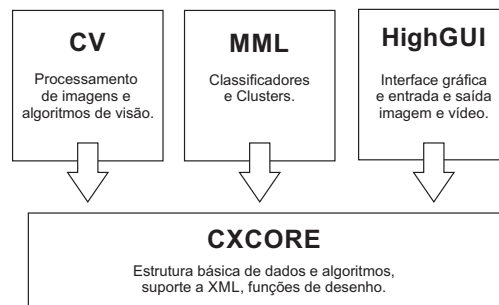


Figura 5.2: Estrutura básica do *OpenCV* adaptado de [4]

Na Figura 5.2 é visto a estrutura básica do *OpenCV*; ou seja, os módulos da biblioteca onde:

- CV - Contém componentes de visão de baixo nível para processamento de imagens e visão computacional;
- MML - Parte relacionada a aprendizagem de máquina e inclui classificadores estatísticos e ferramentas de clusters;
- HighGUI - é relacionado as rotinas de entrada e saída e funções para carregamento de vídeos e imagens;
- CxCORE - Dispõe das estruturas de dados básicas.

- CvAux - Contém algoritmos experimentais, mais não foi referenciado na Figura da estrutura base.

5.3 Etapa 1: Aquisição da imagem

A primeira etapa da solução proposta é dada pela aquisição das imagens que contém parasitos do tipo *plasmodium*, onde um dos grandes desafios é relacionado ao equipamento para realizar essa operação, pois os parasitos do tipo *plasmodium* são pequenos e necessitam de um equipamento de microscopia com ampliação de no mínimo 1000X.

Alguns pré-requisitos foram estabelecidos como forma de padronização a esta proposta, conforme segue:

1. Equipamento de microscopia com no mínimo 1000X de ampliação dado pela junção de lentes objetivas de 100X , lentes oculares de 10X e óleo de imersão, justifica-se este item pelo fato de ser o requisito mínimo para a indentificação e parasistos segundo [21];
2. Equipamento de captura de imagens: máquina fotográfica, dispositivo móvel (tablet e celular), webcam USB conectada a um dispositivo móvel (Notebook). Justifica-se este item por ser uma proposta de baixo custo e pelo fato de utilizar dispositivos móveis.
3. Sistema de acoplamento caseiro de dispositivos e câmeras. Justifica-se este item pela proposta de obter um baixo custo no geral;

Justifica-se uma proposta de baixo custo; dispositivos de captura de imagens com valor igual ou inferior a R\$ 400,00; sistema de acoplamento da câmera ao microscópio não ultrapassar R\$ 50,00; dispositivos móveis para uso do protótipo com o custo igual ou inferior à R\$ 400,00. No geral os custos com exceção do microscópio não devem ultrapassar R\$ 850,00.

Em seguida será detalhado os itens da fase de aquisição da imagem:

Item 1 - Equipamento de microscopia

O equipamento de microscopia laboratorial foi disponibilizado pelo Centro de Pesquisas Aggeu Magalhães (CPQaM). Dentre suas características, ele possui lente objetiva de 100X e lente ocular de 10X , o que totaliza 100 vezes a ampliação. Por sua vez, este equipamento

não possui câmera. Para tal, foi proposto um equipamento de captura de imagens que será visto nos próximos tópicos. Infelizmente não foi possível testar com outros equipamentos de microscopia, haja visto, a dificuldade de acesso a esses equipamentos.

Item 2 - Equipamento de captura de imagens

Primeiramente, foram analisados três equipamentos de captura de imagens: webcam de alta resolução; uma câmera fotográfica com resolução de 5 MP, um tablet com resolução de 3.2 MP.

Iniciando pela webcam o seguinte equipamentos foi analisado: Logitech c270 com resolução de 1280x720 pixels e sensor de 3MP, ilustrado na Figura 5.3.



Figura 5.3: Webcam logitech c270

A câmera foi acoplada à lente ocular do microscópio bacteriológico, com o objetivo de obter imagens de parasitos do tipo *plasmodium*. Foi utilizado um sistema de acoplamento caseiro (será visto no item 3 desta seção). Com isso, não obtive alguma intervenção ou alteração por parte de hardware.

O outro equipamento testado foi uma câmera digital Sony DSC H1, que possui um sensor de 5MP e um de zoom ótico de até 12X, ou seja, caso houve-se necessidade de utilizar o zoom ótico para aproximar e/ou ampliar o tamanho das células sanguíneas, não haveria perda de qualidade ou de pixels. Porém no experimento não foi possível utilizar o zoom ótico, pois com a ampliação tornou-se inviável visualizar os parasitos, exemplificando em um campo de visão normal sem o zoom ótico era possível visualizar cerca de 30 células sanguíneas, e com o zoom habilitado este número reduziria para 2 ou 3 células, o que inviabiliza o uso desse recurso devido a pequena quantidade de amostras a serem geradas. A câmera é ilustrada na Figura 5.4.



Figura 5.4: Camera Sony DSC H1

O terceiro equipamento analisado é um dispositivo móvel com o Sistema Operacional *Android* 4.1[8] que dispõe de uma camera traseira com a resolução de 3.2MP. Trata-se do *Samsung galaxy tab 2*. Conforme ilustrado na Figura 5.5



Figura 5.5: Samsung galaxy tab 2

Na Tabela 5.1 são dadas as especificações dos equipamentos.

	Sony DSC H1	Logitech c270	Samsung Galaxy TAB2
Dimensões (mm)	71 x 30 x 91	0.8 x 2.1 x 1.7	175 x 257 x 9.7
Peso (g)	591	100	587
Resolução da Câmera (MP)	5MP	3MP	3.2MP

Tabela 5.1: Especificações dos equipamentos.

Item 3 - Sistema de acoplamento experimental

Após definir o equipamento de microscopia e o de captura de imagens faz-se necessário realizar um acoplamento experimental para que estes equipamentos sejam fixados a lente ocular do microscópio sem danificá-lo, e como isto realizar a captura das imagens de *plasmodium*. Devido às dimensões e peso, o equipamento *Sony DSC H1* não obteve êxito no acoplamento experimental.

Para acoplar a webcam foi realizada uma busca por alguma peça pré-moldada que obtivesse características de fixação na ocular do microscópio bacteriológico. Assim, uma redução hidráulica de canos de 20mm para canos de 40mm foi necessária. Foi realizada um corte, lixamento e uma pintura na peça, conforme Figura 5.6.

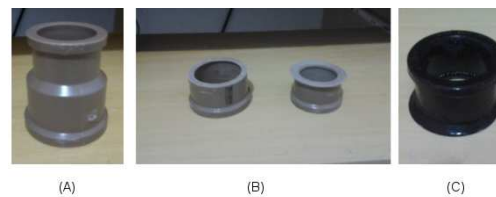
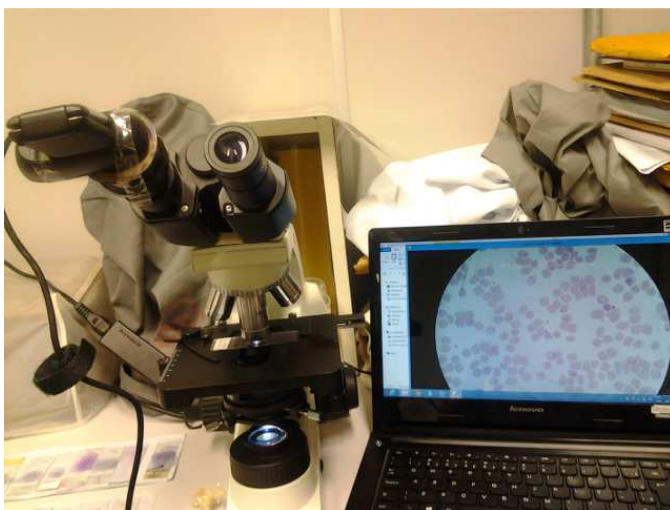
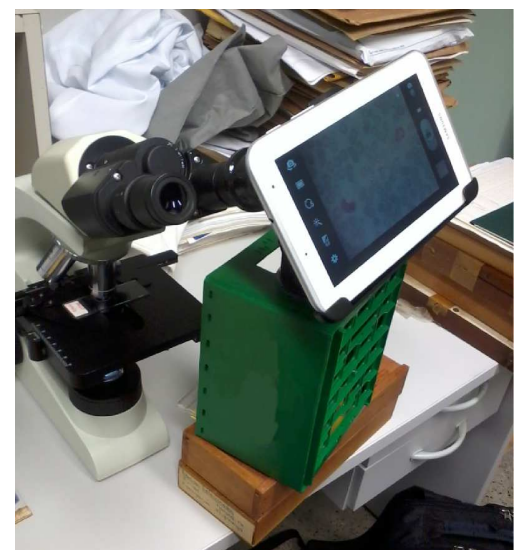


Figura 5.6: Acoplamento experimental utilizando redução hidráulica, onde: (A)Corresponde a peça original, (B)Corte e lixamento e (C)Peça concluída.

No acoplamento do tablet ao microscópio foi necessário, além do acoplamento experimental, um suporte extra, sendo este é um equipamento geralmente utilizado em veículos automotores para fixar dispositivos móveis ao vidro dianteiro dos veículos. Para este projeto, o suporte foi adaptado sobre caixotes , onde a altura ideal depende das dimensões do microscópio. A Figura 5.7 ilustra os acoplamentos.



(a) Webcam adaptada



(b) Tablet adaptado

Figura 5.7: Acoplamento dos equipamentos ao microscópio

Com base nos itens observados, foi possível realizar a adaptação com peças e equipamentos de baixo custo financeiro. O acoplamento com o sistema experimental foi proposto

por [15] e mostrou-se eficiente no quesito custo benefício. Conforme visto anteriormente, a adaptação não foi realizada com êxito na câmera Sony DSC H1. O acoplamento do Webcam é básico, necessitando apenas de ajuste no foco e na distancia do sensor da câmera à lente do microscópio, este pode ser realizado apenas afastando ou retraindo a peça. Para o dispositivo móvel, o uso de caixotes pode ser substituído por qualquer item que tenha uma base fixa e que se ajuste a altura do microscópio.

Esta etapa de aquisição de imagens consistiu em testar três equipamentos para captura de imagens de parasitos de malária. Para a realização dos testes foram utilizadas lâminas de esfregaço sanguíneo na etapa de *trofozoítos*, fase inicial do ciclo da malária e ilustrada na Figura 2.2 no Capítulo 2. Dentre os três equipamentos, nenhum obteve uma excelente qualidade na captura, justificado pelo fato de que nessa etapa do ciclo da malária o parasito apresenta forma anelar, onde qualquer desfoque na câmera pode prejudicar na captura do parasito. Devido a característica de baixo custo do projeto, não houveram testes com equipamentos (câmeras) profissionais, o que de fato poderia acarretar em uma melhora significativa na captura das imagens.

5.4 Etapa 2: Pré-processamento e segmentação

Os processos de visão computacional na sua grande maioria necessitam de uma etapa de pré-processamento envolvendo a área de processamento de imagens, onde é necessário padronizar tais imagens em um determinado formato e aplicar filtros a fim de remover algum ruído que por ventura tenha sido criado na etapa de aquisição. A área de processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio do computador. A segmentação de imagens área da visão computacional consiste em particionar a imagem em regiões[26].

Para o modelo proposto foi definido um fluxo de etapas com o objetivo de reduzir o espaço de busca relacionado à indentificação dos parasitos dentro das células, ilustrado na Figura 5.8:

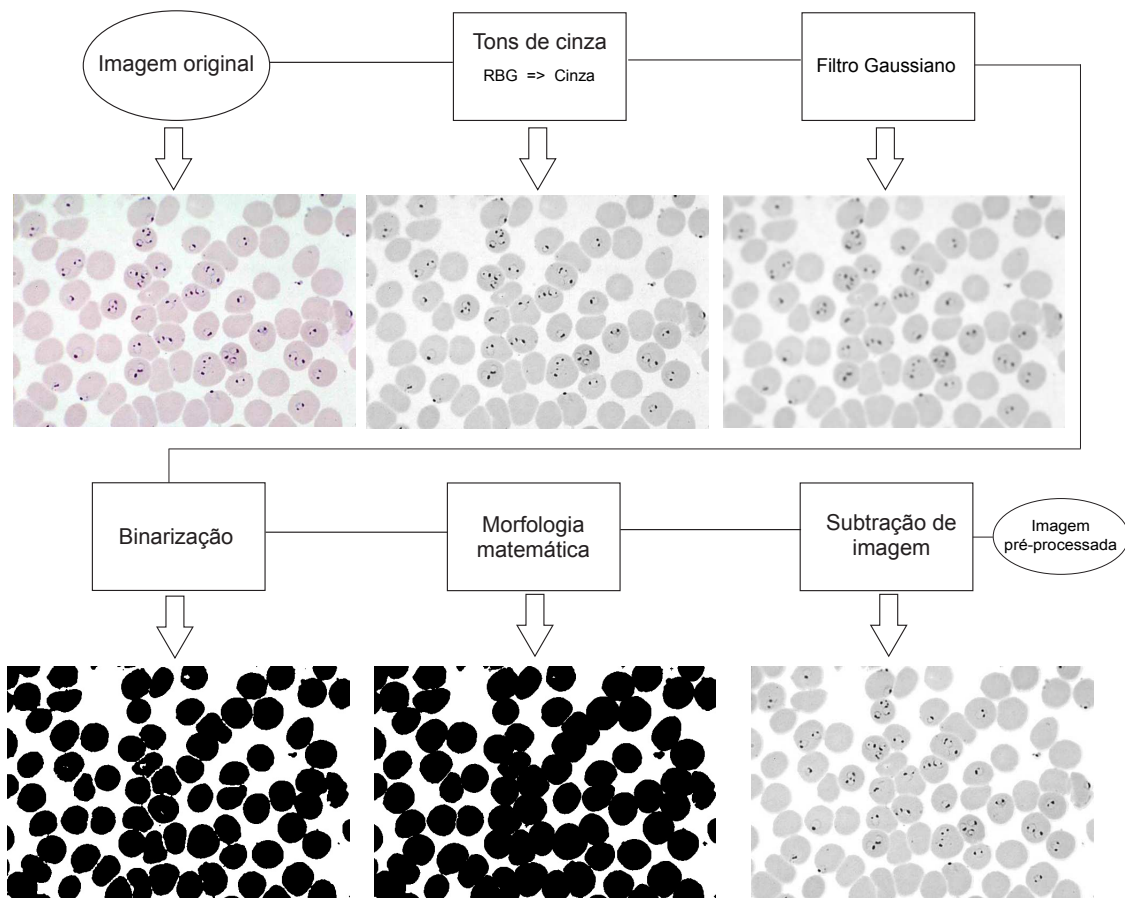


Figura 5.8: Fluxo Geral

Discorrendo sobre o fluxo apresentado na Figura 5.8, inicialmente é dada uma imagem em cores *RGB*, onde podemos denominar como imagem original. Este modelo foi utilizado por ser o mais popular e de simples implementação, ele é utilizado em dispositivos como monitores e câmeras de vídeo, desta forma, não necessitando conversões para outros modelos de cor. Esta contém uma imagem adquirida através do método de esfregaço sanguíneo (visto no Capítulo 2). No geral, a ideia dessa etapa de pré-processamento é reduzir o espaço de busca; ou seja, separar as células sanguíneas do fundo. Para tal operação não é necessário utilizar todo o espaço de cores do modelo *RGB*, visto que realizar operações de segmentação em imagens com várias camadas de cores pode ser bastante custoso. Portanto a imagem original é transformada para tons de cinza vide Equação 5.1 segundo [33].

$$res(x, y) = R(0.299) + G(0.587) + B(0.114) \quad (5.1)$$

onde, $res(x, y)$ é imagem em tons de cinza resultante dessa operação. Após a conversão a imagem passa a ter apenas uma única amostra no espaço de cores variando entre preto de menor intensidade e branco de maior intensidade[2].

Uma das técnicas mais utilizadas na área de processamento de imagens é o processo de filtragem, onde algoritmos são aplicados a fim de eliminar regiões indesejáveis. Existem dois grupos os filtros: Lineares e Não Lineares. Esse processo de filtragem é realizado utilizando matrizes que são chamadas de máscaras conhecidos também como convolução e são aplicadas nas imagens. Nesse sentido, onde os filtros de média e gaussianos são amplamente utilizados para eliminar ruídos [26]. O tipo mais comum são os filtros lineares, em que o valor do pixel de saída $g(i, j)$ é determinado pela soma ponderada dos valores de pixels de entrada $f(i + k, j + l)$:

$$g(i, j) = \sum_{k,l} f(i + k, j + l)h(k, l) \quad (5.2)$$

onde, $h(k, l)$ é o *kernel*, ou seja, os coeficientes do filtro, isso determina os filtros como uma janela de coeficientes ao longo da imagem. Dando continuidade ao fluxo proposto, é utilizado nesse projeto um filtro gaussiano que segundo [5] pode ser bastante útil mesmo não sendo o mais rápido em processamento. Este filtro é dado pela colovução de cada ponto da matriz de entrada aplicado com um *kernel* gaussiano; ou seja, uma função gaussiana, onde é dado uma matriz de saída que é o resultado do filtro[5]. Foi utilizado um *kernel* de 7×7 do filtro gaussiano, onde o resultado dessa operação é uma imagem suavizada. Justifica-se essa operação para uma melhor extração de *background* ou fundo, pois com a redução de detalhes, alguns elementos da imagens tais como as bordas das células obterão destaque, é possível verificar que os parasitos de *plasmodium* contidos na imagem perdem algumas características, porém não serão necessários nessa etapa.

Na área de visão computacional existe um processo denominado limiarização ou binarização, onde em sua concepção busca separar grupos de cinza em uma determinada imagem, ou seja, transformar essa imagem em apenas dois grupos da escala de cinza 0(100% *preto*) e 255(100% *branco*). Porém, existe um complicador nesse processo, que consiste em determinar o limiar ou ponto de corte ideal afim de separar esses dois grupos. Como exemplo, dado uma determinada imagem em tons de cinza e um limiar 127, pode-se ter uma operação que todos os *pixels* que forem menor que esse valor passa a ser de uma classe e o que for maior de uma outra classe. Definir esse ponto de corte é o grande desafio dessa etapa, uma vez que cada imagem possui características diversas, dado que as imagens foram capturadas por dispositivos de captura de vídeo e câmeras, onde podem ocorrer ruídos ou interferências eletromagnéticas.

Na Seção 5.3 foram definidos os equipamentos de captura de imagens de *plasmodium*, per-

mitindo buscar um algoritmo de binarização. Para tal, a técnica de limiarização *OTSU*[34] foi escolhida por ser amplamente utilizada na literatura e também pela característica do ponto de corte(limiar) automático, uma vez que não foi possível encontrar um modelo para extrair células do fundo da imagem [26]. A técnica *OTSU* propõe uma abordagem não paramétrica (não estima os parâmetros do modelo) e não supervisionada para encontrar o limiar automático em uma dada imagem[34]. O método *OTSU*[34], basicamente, assume que a imagem contém duas classes de *pixels*, ou seja, objeto e fundo, representados respectivamente por $C1$ e $C2$. Nesta etapa, ele transforma a imagem proveniente da suavização por filtro gaussiano em uma imagem binária, contendo apenas duas classes: objeto, que neste caso são as células sanguíneas e *background*, que é todo o restante da imagem. Para o caso da detecção de malária, o *background* não é importante, uma vez que a maior parte dos parasitos encontra-se dentro das células, no caso das lamínas preparadas para o esfregaço sanguíneo[24].

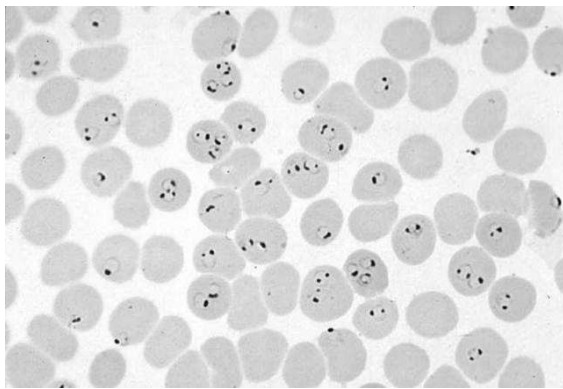
Após a etapa de limiarização ou binarização, é necessário uma nova etapa, pois para este caso em específico de detecção de malária pode-se notar que alguns *pixels* foram limiarizados erroneamente dentro das células sanguíneas, áreas que, na verdade, pertencem às células mas foram definidas como fundo, conforme visto na Figura 5.8. Para diminuir ou até buscar uma melhor classificação desses *pixels*, é dada uma nova etapa que busca utilizar técnicas de morfologia matemática para tal operação.

Segundo Pedrine [2] a morfologia matemática consiste em uma metodologia de análise de imagens que permite a construção de operadores úteis à descrição de objetos em imagens digitais, podendo ser utilizados em uma grande quantidade de aplicações em processamento de imagens, tais como: redução de ruído, isolar elementos individuais, busca de padrões individuais, detecção de bordas, ou até simplesmente detectar uma área irregular na imagem. Como exemplo, para este projeto, após a binarização, alguns *pixels* dentro das células foram marcados erroneamente necessitando de uma operação para diminuir esses ruídos. A morfologia matemática consiste basicamente de um conjunto de pixels que são reunidos em grupos formando uma estrutura bidimensional $f(x, y)$, utilizando-se a teoria de conjuntos matemáticos, que consiste em extrair informações geométricas e de topologia de conjuntos não definidos (imagem a ser processada) através de outro conjunto bem-definido, chamado elemento estruturante (máscara aplicada sobre a imagem).

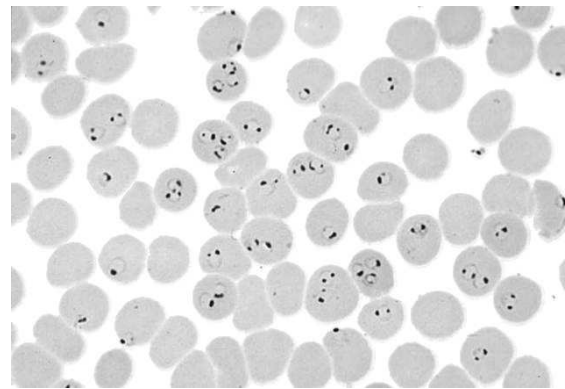
Dando continuidade ao processo de pré-processamento deste projeto, realiza-se uma

operação morfológica de erosão nas imagens após a limiarização. A erosão busca combinar dois conjuntos usando vetores de subtração, onde é dada a interseção dos conjuntos A e B. Assim, $A \oplus B = \{z | (B)_z \subseteq A\}$ [26].

Com a finalização dessa etapa de remoção de fundo, a imagem passou por diversas transformações a fim de localizar as células sanguíneas e com isso restringir o espaço de busca do algoritmo de classificação que será visto mais adiante para estas células. Ao ter a imagem resultante da erosão, é realizada uma subtração com a imagem convertida a níveis de cinza e com isso o fundo ao redor das células é removido. A Figura 5.9 demonstra essa remoção de fundo e o final da etapa de pré-processamento.



(a) Imagem *padrão ouro* em tons de cinza.



(b) Imagem com remoção de fundo.

Figura 5.9: Resultado da remoção de fundo

A Figura 5.9a representa a imagem em tons de cinza com o background e a Figura 5.9b já mostra a imagem resultante das operações anteriores já com a remoção de fundo concluída.

Ao fim dessa etapa houve uma melhora visual na qualidade da imagem, uma vez que, na área de pré-processamento de imagens e visão computacional essa etapa é fundamental para encontrar-se objetos e até mesmo restringir o espaço de buscas. Existem diversos exemplos de aplicações como, por exemplo, na identificação de placas em automóveis, onde é necessário primeiro realizar a segmentação de toda imagem afim de descobrir a localização da placa, para em seguida realizar as operações para reconhecimento de caracteres e padrões[26]. Esse processo não diferencia do nosso propósito que, nessa etapa, é de separar o fundo dos objetos de interesse que, em nosso caso são as células sanguíneas em uma primeira etapa, para em seguida fazer a identificação de parasitos no interior dessas células. Na próxima seção, será visto a etapa de reconhecimento de padrões.

5.5 Etapa 3: Treinamento e validação

Com a conclusão das etapas anteriores, segue-se duas novas etapas: treinamento e classificação de imagens.

O treinamento foi realizado utilizando o método de detecção de faces proposto por Viola e Jones[3] e descrito no Capítulo 4 por já terem trabalhos com resultados significativos relacionados ao reconhecimento e diagnóstico de outras doenças [15][35][36]. A implementação do treinamento foi feito nas linguagens de programação C/C++ e Python, utilizando a biblioteca de visão computacional *OpenCV*[32]. Algumas atividades foram definidas para esta etapa:

- Criar conjunto de exemplos positivos e negativos;
- Treinar o classificador com os conjuntos de exemplos.

5.5.1 Atividade 1: Criar o conjunto de exemplo positivos e negativos.

As imagens foram adquiridas através de uma aplicação desenvolvida especificamente para realizar essa captura utilizando a linguagem de programação *Python* e a biblioteca de visão computacional *OpenCV*[32]. Abaixo é ilustrado o pseudocódigo.

Algoritmo 4 Capturar imagens da camera

```
1: Inicializa camera
2: enquanto Verdadeiro faça
3:   Exibir imagem da camera
4:   Capturar entrada do teclado
5:   se Pressionar o botão 0 então
6:     Salva a imagem no diretório de negativas
7:   fim se
8:   se Pressionar o botão 1 então
9:     Salva a imagem no diretório de positivas
10:  fim se
11: fim enquanto
```

No Algoritmo 4, primeiramente é selecionado o equipamento de captura, neste caso, a webcam. Logo após, é dada uma estrutura composta por exemplos positivos e negativos

e ao pressionar um evento no teclado as imagens são salvas nos respectivos diretórios, sem qualquer etapa de pré-processamento. Este algoritmo proposto foi utilizado para o caso de captura de imagens com uma webcam acoplada, conforme visto anteriormente neste capítulo. Para os demais equipamentos de captura, as imagens foram capturadas e guardadas em respectivos repositórios, contendo imagens positivas e negativas, de forma manual sem a necessidade de tal aplicação.

Após a captura das imagens é necessário marcar os objetos de interesse, no caso os parasitos de tipo *plasmodium*, para tal o algoritmo 5 é proposto.

Algoritmo 5 Marcar objetos

```

1: Entrada: Imagens positivas contendo parasitos
2: Saída: Localização  $f(x, y)$  de parasitos para cada imagem

3: Carrega imagens positivas
4: Habilita cursor do mouse                                ▷ Selecionar o objeto
5: enquanto Verdadeiro faça
6:     Exibe imagem positiva
7:     se Cursor do mouse selecionar objeto então
8:         Salvar coordenadas  $f(x, y)$ 
9:         Marcar objeto
10:    fim se
11:    se Pressionar Espaço então
12:        Salva coordenadas  $f(x, y)$  do objeto e largura e altura
13:    fim se
14:    se Pressionar Enter então
15:        Passa para a próxima imagem
16:    fim se
17:    se Pressionar escape então
18:        Finaliza o laço e fecha o programa
19:        Salva arquivo em texto os dados ▷ Salva em arquivo de texto para cada imagem a quantidade de
    objetos as respectivas coordenadas  $f(x, y)$  dos parasitos bem como largura e altura.
20:    fim se
21: fim enquanto

```

O algoritmo para marcar os objetos tem a finalidade de salvar a localização dos parasitos para cada imagem, um exemplo de marcação de um objeto pode ser visualizado na Figura 5.10, o arquivo de saída deste programa é um arquivo no formato de texto contendo as seguintes informações: nome do arquivo, quantidade de objetos, posição x e y, altura e largura.

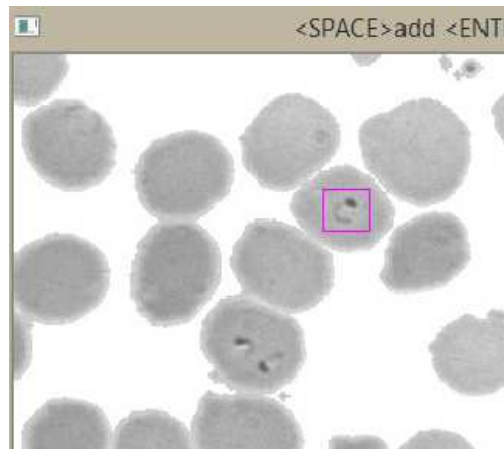


Figura 5.10: Software de Marcação de Objetos.

Após o arquivo ser criado com todas as amostras positivas é necessário criar um arquivo contendo a localização das imagens negativas. Este é um arquivo de texto onde consta somente a localização dessas imagens, a escolha das imagens negativas é crucial para o desempenho do classificador, Viola e Jones [3] sugere o uso de imagens com características variadas para detecção de faces, como por exemplo, pontes, ruas, casas, dentre outros elementos que não se assemelhem as características faciais de um ser humano. Para este projeto, é utilizado um conjunto de imagens negativas de uma fonte proprietária, aliada as imagens adquiridas pelos dispositivos de aquisição descritos no início deste Capítulo. Essas imagens contém células sanguíneas, elementos do sangue e diversos tipos de infecções, quatro exemplos de imagens negativa são ilustrados na Figura 5.11.

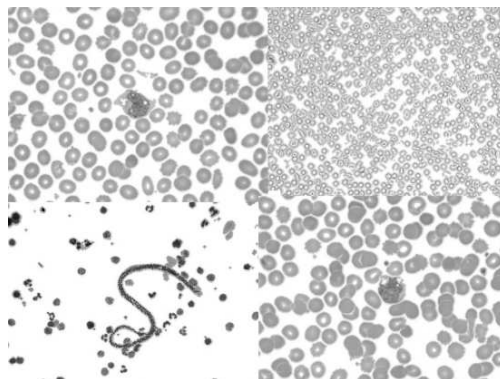


Figura 5.11: Exemplos de imagens negativas.

De posse dos arquivos de textos contendo as imagens positivas e negativas, cria-se um arquivo de amostras. Este tem o objetivo de padronizar as imagens e organizá-las. Para isso

foi utilizada a ferramenta *Createsamples* disponível no OpenCV [32], as imagens positivas são redimensionadas para um tamanho pré-determinado, no caso de Viola e Jones [3] foi definido um tamanho para o caso de detecção de faces 24 X 24 pixels. Para este projeto, foi mantido essas medidas. O resultado dessa operação é um arquivo no formato *.vec* contendo os exemplos já padronizados e prontos para o treinamento. Lienhart [37] relata que com 5000 imagens positivas e 3000 negativas pode-se obter um treinamento satisfatório.

5.5.2 Atividade 2: Treinar o classificador com os conjuntos de exemplos.

A próxima atividade é o treinamento da cascata de classificadores proposta por Viola e Jones [3] e detalhada no Capítulo 4. Lienhart [37] relata que em uma amostra com a subjanela de tamanho 20 X 20 pixels e a quantidade de nós da árvore igual a 2(dois) o classificador tende a ter uma boa taxa de acerto em torno de $0.999^{20} \approx 0,98$ e falsos alarmes $0,5^{20} \approx 9,6e - 07$. De acordo com Viola e Jones [3] o número de estágios da cascata é um fator crucial para o desempenho do sistema de detecção, e uma cascata com 10 estágios já tende a desenvolver uma taxa de acerto de 90% ou $0.990^{20} \approx 0,81$, desde que a quantidade de exemplos seja de 5000 imagens positivas e 10000 imagens negativas.

Para o treinamento desse projeto foi utilizado a ferramenta *haartraining* disponível na biblioteca OpenCV [32]. Somente como uma observação com relação ao treinamento por essa ferramenta, ele pode terminar em um estágio intermediário em algumas condições, por exemplo, quando você atingiu a taxa de acerto mínima desejada ou a taxa de falsos alarmes ou então quando todas as amostras negativas são rejeitadas, isso significa que o número de amostras negativas é insuficiente. Abaixo é dado um exemplo dos parâmetros de um treinamento realizado pela ferramenta *haartraining*:

```
haartraining -data data/UPCandCDC/GAB/90_10/cascadeBasic -vec data/vector_upCdc_90_10.vec -bg negative777/negative.txt -npos 500 -nneg 777 -nstages 20 -mem 1500 -mode BASIC -minhitrate 0.995 -maxfalsealarm 0.5 -w 24 -h 24 -sym
```

onde, *data* é o local onde serão salvos os dados das cascatas, *vec* é a localização do arquivo *.vec* criado no capítulo anterior, *bg* informa a localização de um arquivo de texto contendo o caminho para as imagens negativas, *-npos* é o número de exemplos positivos

disponibilizados no arquivo *.vec*, *-nneg* o total de exemplos negativos localizados no arquivo *.txt*, *mem* a quantidade de memória disponibilizada para o treinamento. De acordo com [32] o padrão é de apenas 256MB, o número máximo a ser especificado seria de 2GB, pois há um limite em uma CPU de 32bits ($2^{32} \approx 4GB$), *mode BASIC* está relacionado às características de *haar* utilizadas para o treinamento, onde *BASIC* são apenas as características definidas por [3] e *ALL* para as características definidas por [37]. Para este projeto o modo *BASIC* mostrou-se mais eficiente, *minhitrate* delimita a taxa de acerto mínima para que um estágio da cascata seja concluído, *maxfalsealarm* é dado para a taxa mínima de falsos, alarmes; quanto menor essa taxa mais preciso será o classificador, *w* e *h* respectivamente a largura e altura da subjanela, *-sym* é um parâmetro que é visto se há simetria ou não na imagem.

O resultado do treinamento é um arquivo no formato *eXtensible Markup Language* (.xml), contendo uma árvore de decisão binária.

5.6 Etapa 4: Desenvolvimento de um sistema de detecção de *plasmodium* para dispositivos móveis.

Nesta seção, é relatado o desenvolvimento de uma solução capaz de detectar parasitos de *plasmodium* em dispositivos móveis, sem a necessidade de intervenção de *hardware*.

Existem atualmente três grandes desenvolvedores de sistemas operacionais para dispositivos móveis, são eles: Google, Apple e Microsoft, respectivamente com seus sistemas: Android, IOS e Windows Phone 8.1 [8][9][38][39].

O Android é o sistema operacional de código aberto mais popular do mundo com aproximadamente 1 bilhão de usuários conectados em diversos dispositivos móveis (tablets, smartphones), é desenvolvido em C/C++ e baseado em *kernel* Linux. O sistema operacional é composto de uma plataforma de desenvolvimento denominada *System Development Kit* (*SDK*), incluindo dentre outras ferramentas: interface de usuário, framework de arquitetura, interação com serviços do google, e diversas bibliotecas e kits de desenvolvimento de forma a facilitar a experiência do desenvolvedor. Os aplicativos para Android em sua grande maioria são desenvolvidos na linguagem de programação Java[40], porém, há uma extensão onde pode-se criar aplicações em C/C++ que é denominada *Native Development Kit* (*NDK*)[9][8].

Para este projeto foi escolhido o Android, pelo fato de ser um sistema operacional livre e disponível em diversos dispositivos, independente do fabricante do hardware, bem como por sua junção entre o SDK e o NDK, disponíveis nesse sistema. A biblioteca de visão computacional OpenCV[32] também tem algoritmos implementados nessa plataforma.

No desenvolvimento do sistema proposto foi necessário o uso das linguagens de programação Java e C/C++, onde, são dadas duas implementações: reconhecimento por através de uma imagem estática e reconhecimento de parasitos através de um acoplamento experimental, fazendo-se o uso de um microscópio bacteriológico. O algoritmo de detecção de faces de Viola e Jones [3] é bastante utilizado para aplicações em tempo real, por suas diversas otimizações como, por exemplo o uso da imagem integral, conforma visto no capítulo 4.

No primeiro experimento foi desenvolvido um aplicativo para detecção de parasitos de *plasmodium*, a partir de uma imagem o fluxo que é ilustrado na Figura 5.12.

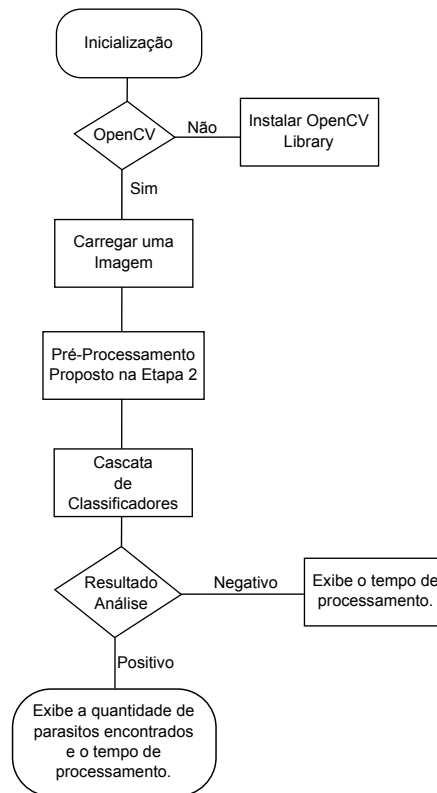


Figura 5.12: Fluxo do aplicativo de detecção por imagem.

onde, a partir da inicialização do aplicativo, verifica-se se há a instalação da biblioteca OpenCV, caso positivo é apresentada a interface do sistema com a opção de carregar uma imagem. Ao pressionar o botão analisar, é realizada a etapa de pré-processamento e poste-

riormente a varredura em subjanelas da cascata de classificadores em busca de parasitos de *plasmodium*. A cascata de classificadores é composta por um arquivo .xml gerado na etapa 3 de Treinamento e Validação. No caso da subjanela é utilizado multi-escala, ou seja, a janela de busca é aumentada em até uma escala de 1.5 . Com o resultado da detecção, é somado o tempo total percorrido e plotados na interface. Uma imagem da interface do sistema pode ser visualizada na Figura 5.13.

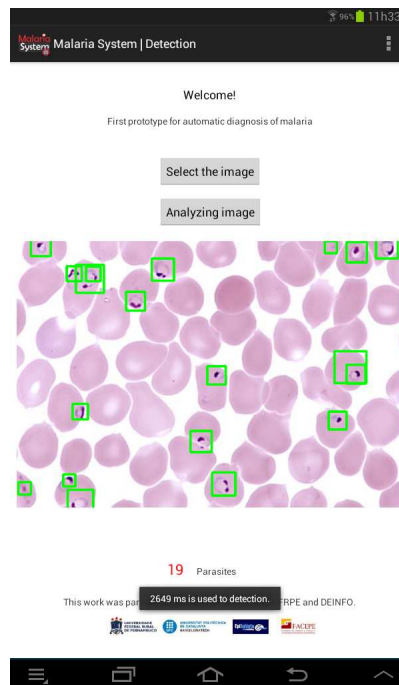


Figura 5.13: Interface do aplicativo de detecção por imagem.

As fases de pré-processamento e cascata de classificadores ilustradas na Figura 5.12 foram desenvolvidas em C/C++ através do NDK. As outras etapas utilizaram o padrão do Android o Kit SDK.

O segundo experimento é dado por um aplicativo de detecção de plasmodium em tempo real, essa operação pode ser realizada através do acoplamento experimental conforme visto na Etapa 1 de Aquisição da Imagem. No Algoritmo 6 é dado o pseudocódigo desse experimento.

Algoritmo 6 Sistema de detecção de *plasmodium Android*

- 1: Inicializa camera
- 2: Define a camera para 640 X 480 pixels
- 3: **enquanto** *Verdadeiro* **faça**
- 4: Exibir imagem da camera
- 5: Executa Varredura da imagem através de sub-janelas de 24 x 24 pixels
- 6: **se** *Encontra o objeto de interesse* **então**
- 7: Marca o a área do objeto encontrado
- 8: **fim se**
- 9: **Se não** Continua procurando
- 10: **fim enquanto**
- 11: **se** *Precionar o botão return* **então**
- 12: Encerra o aplicativo e libera a câmera
- 13: **fim se**

Nesse experimento, ao invés do carregamento de uma imagem, utiliza-se a câmera nativa do equipamento, para realizar detecção através da câmera traseira do dispositivo móvel, operando a uma resolução de 640 X 480 *Pixels*, obtendo-se uma taxa em média de 5 quadros por segundo (QPS) . O aplicativo pode ser utilizado acoplado a um microscópio bacteriológico. Sua taxa de detecção leva em conta diversos fatores, tais como: distância, iluminação, e o conjunto de treinamento. A câmera é fixa e a atualização do campo de visão é realizada normalmente no microscópio bacteriológico. A detecção dos parasitos é realizada frame a frame, conforme proposta de [3].

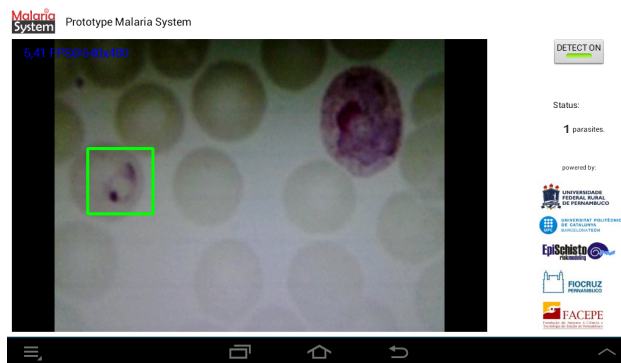


Figura 5.14: Sistema de detecção no dispositivo móvel.

Em ambos os aplicativos gerados com essa dissertação é utilizado o melhor classificador do treinamento que será visto no próximo Capítulo. No entanto já mostra-se possível a implementação de um software de baixo custo a um dispositivo móvel para a detecção de parasitos do tipo *plasmodium*.

Capítulo 6

Resultados

Neste capítulo, serão apresentados os resultados obtidos com relação à classificação das imagens de malária em casos: positivos, falsos positivos, negativos e falso negativos, através do *framework* do *MalariaSystem*, ilustrado na Figura 5.1 no Capítulo 5. Serão avaliados 4 conjuntos de dados distintos. Posteriormente, a junção de todos estes conjuntos também será avaliada. Além desses resultados, serão apresentados também testes com os melhores conjuntos de treinamento apresentados em uma implementação para dispositivo móvel, e com isso poder avaliar a sua viabilidade.

Para realizar os experimentos a seguir foi utilizado um computador com as seguintes configurações: processador Intel(R) Core (TM) i3-3717U (1,80GHz), 4,00GB memória RAM, e sistema operacional Microsoft Windows 8.1, bem como alguns parâmetros foram estabelecidos no *Haartraining* como padrão a todas as amostras do conjunto de treinamento, conforme Tabela 6.1.

Parâmetros <i>haartraining</i>					
Mem	Min hit rate	Max false alarm rate	Mode	Width	Height
512	0.995000	0.500000	Basic	24	24

Tabela 6.1: Características padrão a todos os conjuntos de treinamento.

onde, *Mem* especifica a quantidade de memória utilizada no treinamento; *Min hit rate* especificação de uma taxa de acerto mínima, *Max false alarm rate* é dado como o alarme falso máximo aceito pelo sistema; *Mode* características de *haar* utilizadas, descritas na Figura 4.2 do Capítulo de Fundamentos do ShistoSystem e propostas por Lienhart[30]; *Width* largura da sub-janela ; *Height* altura da sub-janela.

6.1 Experimentos

Os experimentos tem o propósito de avaliar a qualidade e a viabilidade dos resultados obtidos para detecção de parasitos de malária em dispositivos móveis, ou seja, quantificar a taxa de acertos. A avaliação será somente em duas classes, respectivamente: positivos (infectado por parasitos do tipo *plasmodium*) e negativos (não infectados). Para tal, a Tabela 6.2 ilustra as nomenclaturas utilizadas.

Siglas	Significado
P	Positivos
N	Negativos
VP	parasitos corretamente detectados como positivos
VN	Objetos corretamente detectados como negativos
FN	Parasitos positivos não detectados pelo classificador
FP	Objetos que não são parasitos do tipo <i>plasmodium</i>

Tabela 6.2: Siglas utilizadas para avaliação.

Na área de inteligência artificial existem algumas métricas para avaliação de desempenho. Com base em alguns trabalhos da literatura [41] [42] [43] são usados duas métricas afins de avaliação de classificadores, são elas; *sensitividade (recall)* e *especificidade*, onde respectivamente mede a porcentagem de amostras positivas classificadas corretamente sobre

o total de amostras positivas e a outra a porcentagem de amostras negativas identificadas corretamente sobre o total e amostras negativas, vide equações 6.1 e 6.2.

$$recall = \frac{VP}{VP + FN} \quad (6.1)$$

$$especificidade = \frac{VN}{VN + FP} \quad (6.2)$$

Nas próximas seções serão vistos os resultados dos experimentos de acordo com cada base de imagens, bem como um treinamento geral contendo todas as imagens.

6.1.1 Experimento 1

A primeira base de imagens é composta por exemplos positivos disponibilizado pelo grupo de pesquisas *Discrete Modelling and Simulation of Biological Systems* (MOSIMBIO) da Universidade Politécnica da Catalunha (UPC) aliadas às imagens de um instituto inglês que armazena uma biblioteca com o maior catálogo de imagens médicas, manuscritos e ilustrações ligados à ciência do mundo, denominado *Wellcome Images* [44][45]. Na Tabela 6.3 são plotados os dados referentes à divisão da base de imagens entre treinamento e validação. Desta forma, os resultados dos experimentos foram gerados a partir da base de validação, a escolha dessas imagens foi feita de forma aleatória. Todas as imagens tem características distintas dentre elas: diferença de luminosidade, equipamento de aquisição.

Positivas	Negativas	Validação	Porcentagem Treinamento / Testes	Total Positivas
340	777	217	70% 30%	557
446	777	111	80% 20%	557
500	777	57	90% 10%	557
523	777	34	94% 6%	557

Tabela 6.3: Dados de entrada do experimento 1

Para o primeiro cenário é dado um conjunto de imagens positivas com 70% das imagens reservadas para o treinamento e 30% para testes, os experimentos tiveram os resultados plotados na Tabela 6.4.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	109	1655	108	1315	217	0,51	0,37	55%
15	117	466	100	466	217	0,53	0,62	61%
20	98	165	119	165	217	0,45	0,82	67%

Tabela 6.4: Resultados para o cenário 70% treinamento e 30% validação- experimento 1

De acordo com os resultados na tabela 6.4 é possível observar que o classificador com 20 estágios obteve um melhor desempenho. Isto se refere à taxa de acerto e a especificidade. Foi utilizado um conjunto de testes com 217 exemplos positivos.

Para um segundo cenário é proposto um conjunto com 80% para treinamento e 20% para testes, os resultados estão na Tabela 6.5.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	54	1013	57	567	111	0,48	0,57	63%
15	61	676	50	230	111	0,54	0,77	72%
20	45	527	66	81	111	0,40	0,90	79%

Tabela 6.5: Resultados para o cenário 80% treinamento e 20% validação- experimento 1

Novamente o classificador com 20 estágios obteve uma melhor taxa de acerto, em relação ao experimento anterior a melhoria relacionada a esta taxa foi de 12%.

No próximo cenário é dado 90% para treinamento e 10% para testes, comparado ao primeiro cenário há um aumento de 20% em relação ao conjunto de treinamento. Vale salientar que os exemplos positivos e negativos disponíveis nesse trabalho correspondem a menos de 10% do valor ideal pela literatura [3]. Os resultados são expressos na Tabela 6.6.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	25	986	32	486	57	0,43	0,61	66%
15	23	668	34	168	57	0,40	0,82	77%
20	21	548	36	48	57	0,36	0,94	87%

Tabela 6.6: Resultados para o cenário 90% treinamento e 10% validação- experimento 1

Neste cenário é possível verificar uma baixa no número de falsos negativos (FN). Com esse dado pode-se inferir que a chance de um parasita positivo ser detectado como positivo aumentou. O classificador com 20 estágios obteve uma melhor taxa de acerto com uma especificidade de 94%.

Para buscar um cenário mais diversificado foi também realizado experimentos com 94% do conjunto para treinamento e apenas 6% para testes, vide Tabela 6.7.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	21	1099	13	576	34	0,61	0,57	65%
15	24	739	10	216	34	0,70	0,78	77%
20	21	577	13	54	34	0,61	0,93	89%

Tabela 6.7: Resultados para o cenário 94% treinamento e 6% validação- experimento 1

No ultimo cenário para o experimento 1 o classificador com 20 estágios obteve os melhores resultados. O número de falsos negativos (FN) é três vezes menor que o cenário anterior, a taxa de acerto é de 89% e a sensibilidade, ou seja, as amostras positivas classificadas corretamente sobre o número de amostras positivas foi de 61%.

6.1.2 Experimento 2

Os experimentos 2, 3 e 4 foram produzidos com os conjuntos de imagens adquiridas no Centro de Pesquisas Aggeu Magalhães por Allisson Dantas e supervisionado por Mércia Arruda¹, onde seus respectivos processos de aquisição e sistema de acoplamento foram de-

¹Mércia Eliane de Arruda - Doutora em Saúde Pública e Pesquisadora titular AIII do Departamento de Imunologia da Fundação Oswaldo Cruz/Recife. Fonte: <http://lattes.cnpq.br/3665762834810409>

talhados no Capítulo 5.

Neste experimento, o treinamento foi realizado com as imagens adquiridas pela câmera Sony DSC H1. Na Tabela 6.8 são mostrados os dados referentes às divisões das bases de treinamento e testes.

Positivas	Negativas	Validação	Porcentagem Treinamento / Testes	Total Positivas
81	777	31	70% 30%	112
91	777	21	80% 20%	112
101	777	11	90% 10%	112

Tabela 6.8: Dados de entrada do experimento 2

Esta base teve uma aquisição experimental. Desta forma as imagens contêm ruídos causados pela justaposição de lentes da câmera digital com a lente objetiva do microscópio bacteriológico, e também por não ter sido possível fixar o equipamento a lente. No primeiro cenário o conjunto de imagens é composto por 70% para treinamento e 30% para validação, o número total de imagens desse conjunto é inferior a 5% do proposto pela literatura[3].

A Tabela 6.9 ilustra o primeiro cenário desse experimento.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	11	101	20	1091	31	0,35	0,41	78%
15	4	108	27	239	31	0,12	0,76	79%
20	2	110	29	39	31	0,06	0,95	80%

Tabela 6.9: Resultados para o cenário 70% treinamento e 30% validação- experimento 2

Neste cenário a taxa de sensibilidade é baixa chegando à apenas 6% no classificador com 20 estágios. Desta forma não se tem uma certeza se o classificador está acertando corretamente as classes preditas como negativas. Portanto, estes classificadores não indicados para um sistema de detecção automática.

No segundo cenário deste experimento visto na tabela 6.10 houve uma piora na sensibilidade e na taxa de acerto comparado ao cenário anterior.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	9	885	12	794	21	0,42	0,49	52%
15	6	297	15	206	21	0,28	0,79	57%
20	1	124	20	33	21	0,4	0,95	70%

Tabela 6.10: Resultados para o cenário 80% treinamento e 20% validação- experimento 2

O próximo cenário é particionado em 90% para treinamento e 10% para validação. Os resultados podem ser vistos na tabela 6.11.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	4	153	7	52	11	0,36	0,93	72%
15	2	105	9	41	11	0,18	0,94	89%
20	1	108	10	7	11	0,9	0,99	86%

Tabela 6.11: Resultados para o cenário 90% treinamento e 10% validação- experimento 2

Diante do ultimo cenário deste experimento, é possível analisar uma baixa taxa de sensibilidade para todos os cenários. Portanto as imagens adquiridas somente por esse dispositivo de aquisição não obtiveram uma boa qualidade na classificação.

6.1.3 Experimento 3

Neste experimento é dado um conjunto de imagens adquirida por um dispositivo de webcam e um sistema de acoplamento experimental, ambos descritos no Capítulo 5. A Tabela 6.12 ilustra a quantidade de amostras adquiridas e utilizadas para o treinamento e validação.

Positivas	Negativas	Validação	Porcentagem Treinamento / Testes	Total Positivas
94	777	41	70% 30%	135
108	777	27	80% 20%	135
121	777	14	90% 10%	135

Tabela 6.12: Dados de entrada do experimento 3

Os experimentos foram divididos em 3 cenários, para o primeiro cenário houve a divisão do conjunto de imagens positivas em 70% para treinamento e 30% para testes. Dados plotados na Tabela 6.13.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	17	677	24	583	41	0,41	0,57	53%
15	7	72	34	72	41	0,17	0,91	62%
20	4	21	37	21	41	0,09	0,97	67%

Tabela 6.13: Resultados para o cenário 70% treinamento e 30% validação- experimento 3

No primeiro cenário com 70% a 30% o classificador de 10 estágios obteve um melhor resultado, relacionado à sensibilidade, porém, o número de falsos alarmes foi bastante alto um indicador não interessante para um sistema de reconhecimento de parasitos uma vez que acontecem diversas marcações na imagem de falsos alarmes o que pode causar erros no diagnóstico.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	12	618	15	510	27	0,44	0,60	54%
15	9	177	10	69	27	0,33	0,91	70%
20	1	120	26	12	27	0,03	0,98	76%

Tabela 6.14: Resultados para o cenário 80% treinamento e 20% validação- experimento 3

Em um segundo cenário ilustrado na Tabela 6.14 o classificador com 20 estágios conseguiu 98% de especificidade, ou seja, está com um bom acerto no conjunto de amostras negativas,

porém, a taxa de sensibilidade é de apenas 3%. Com esses dados é possível verificar que este classificador não é ideal para um sistema de detecção.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	7	631	7	184	14	0,50	0,80	55%
15	6	190	8	25	14	0,42	0,96	71%
20	1	133	13	6	14	0,07	0,99	77%

Tabela 6.15: Resultados para o cenário 90% treinamento e 10% validação- experimento 3

No terceiro e ultimo cenário desse experimento plotado na Tabela 6.15, os dados são parecidos com o do cenário anterior, porém, no classificador de 20% estágios é possível observar um acerto de 99% nas amostras negativas. Nesse experimento 3 foi possível observar uma boa taxa de especificidade, no entanto a taxa de sensibilidade foi abaixo dos 50% para a maioria dos cenários. Desta forma, somente este conjunto de imagens não é indicado para gerar um bom classificador para um sistema de detecção.

6.1.4 Experimento 4

Neste experimento, a aquisição da imagem foi realizada através de um dispositivo móvel (*Tablet*), houve uma peculiaridade. Devido ao pequeno conjunto de imagens adquiridas, os testes nesse experimento serão com uma proporção de 70% para treinamento e 30% para validação e apenas cascatas com 10 e 15 estágios. Na Tabela 6.18 é ilustrado o conjunto de dados do experimento.

Positivas	Negativas	Validação	Porcentagem Treinamento / Testes	Total Positivas
19	415	9	70% 30%	28

Tabela 6.16: Dados de entrada do experimento 4

Os resultados são plotados na Tabela 6.17.

Devido a este experimento ter tido um baixo número de imagens para treinamento, os dados plotados na Tabela 6.17 podem sofrer incorreções no sentido de ser menos de 1%

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	5	68	4	49	9	0,55	0,894397	57%
15	5	373	4	354	9	0,55	0,539662	51%

Tabela 6.17: Resultados - experimento 4

do exigido pela literatura. Desta forma, os dados foram plotados somente para fins de comprovação do experimento. De toda forma o classificador com 10 estágios obteve uma melhor especificidade comparado ao de 15 estágios. Os classificadores desse experimento não são indicados para um sistema de detecção nem para comprovação científica.

6.1.5 Experimento 5

Neste experimento será avaliada a junção de todos os conjuntos de imagens positivas. Diante da variedade e diversos tipos de aquisição de imagens, elas podem conter ruídos já relatados neste capítulo, fato este que pode alterar o desempenho e a qualidade do treinamento indicado pela metodologia. Na Tabela 6.18 são ilustrados os dados relativos a esse experimento.

Positivas	Negativas	Validação	Porcentagem Treinamento / Testes	Total Positivas
472	777	203	70% 30%	675
545	777	130	80% 20%	675
648	777	27	94% 6%	675

Tabela 6.18: Dados de entrada do experimento 5

O primeiro cenário será de 70% para treinamento e 30% para testes, resultados na Tabela 6.19.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	116	2565	87	2093	203	0,57	0,27	55%
15	128	1362	75	890	203	0,63	0,46	60%
20	108	728	95	256	203	0,53	0,75	70%

Tabela 6.19: Resultados para o cenário 70% treinamento e 30% validação- experimento 5

Os resultados do cenário visto na Tabela 6.19 superam os 70% da taxa de acerto e uma sensibilidade acima de 50%, o que já indica uma melhora significativa comparada aos experimentos 2, 3 e 4, mesmo quando esta base é composta por todos os exemplos positivos dos 4 experimentos anteriores. Com isso pode-se notar que mesmo com ruído o aumento no numero de amostras trás consigo uma melhoras nos classificadores. Neste cenário a cascata com 20 estágios mostrou-se com um desempenho melhor quanto a taxa de acerto comparado as de 10 e 15 estágios.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	65	1993	65	1448	130	0,50	0,34	57%
15	78	1167	52	622	130	0,60	0,55	64%
20	75	723	55	178	130	0,57	0,81	77%

Tabela 6.20: Resultados para o cenário 80% treinamento e 20% validação- experimento 5

No segundo cenário visto na Tabela 6.20 houve uma melhora na especificidade e sensibilidade na cascata com 20 estágios, com isso tendo um número de falsos alarmes menor comparado ao cenário anterior. A melhor taxa de acerto foi novamente da cascata com 20 estágios, pode-se dizer que este cenário é mais propício que o anterior pelo simples fato do numero de falsos alarmes serem menor.

Para um próximo cenário foi experimentado a combinação de 94% para treinamento e apenas 6% para testes, no geral de todos os experimentos esse cenário foi o que trouxe um melhor resultado na relação taxa de acerto, sensibilidade e especificidade. A Tabela 6.21 ilustra os resultados.

Estágios	VP (Hits)	VN	FN (Missed)	FP (False)	Validação	Recall Sensitividade	Especificidade	Taxa de Acerto
10	18	1144	9	496	27	0,66	0,61	69%
15	19	814	8	166	27	0,70	0,82	82%
20	16	684	11	36	27	0,59	0,95	93%

Tabela 6.21: Resultados para o cenário 94% treinamento e 6% validação- experimento 5

Neste cenário, em seu melhor resultado a cascata de classificadores com 20 estágios obteve uma taxa de acerto de 93% e uma especificidade de 95%.

6.1.6 Curvas ROC *Receiver Operating Characteristic*

A curvas ROC *Receiver Operating Characteristic* é um método de avaliação de sistemas de diagnóstico ou predição de forma gráfica. Um modelo de classificação tem sua representação através de um espaço chamado ROC, onde é calculado a taxa de verdadeiros positivos e falsos positivos (tpr e fpr). As curvas ROC são bastante úteis quando se tem modelos com classes em que existem uma grande desproporção. No resultados do modelo proposto são dadas classes com essa grande desproporção por isso é necessário tal análise[46]. Na Figura 6.1 será plotado as curvas ROCs dos 5 melhores classificadores de todos os experimentos vistos na subseção anterior.

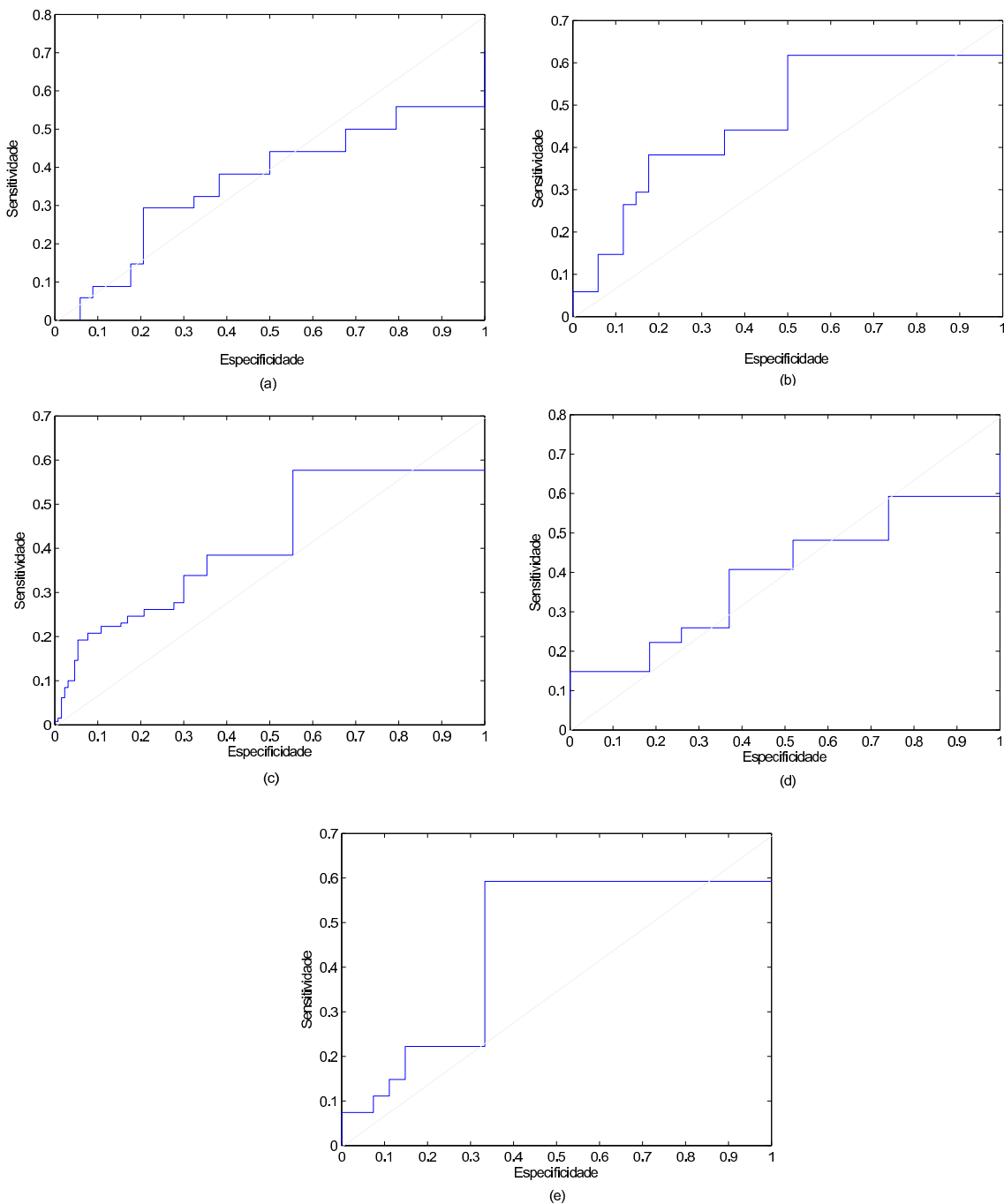


Figura 6.1: Curvas ROC onde são representados os seguintes classificadores : (a)Experimento 1 - 94% treinamento e 6% testes com 15 estágios, (b)Experimento 1 - 94% treinamento e 6% testes com 20 estágios, (c)Experimento 5 - 80% treinamento e 20% testes com 20 estágios, (d)Experimento 5 - 94% treinamento e 6% testes com 15 estágios, (e)Experimento 5 - 94% treinamento e 6% testes com 20 estágios.

A curva ROC plotada na Figura 6.1 (a) apresenta uma média na relação entre sensitivi-

dade e especificidade, neste caso é demonstrado o exemplo de um classificador aleatório, ou seja, não tem uma boa proporção das taxas de especificidade e sensibilidade.

No segundo caso é dado um classificador com uma curva ROC mais acentuada, a área sobre a curva é maior, desta forma tende a caracterizar como um bom classificador. O classificador utilizado na Figura 6.1 (b) pertence ao experimento 1 com 94% treinamento e 6% para testes em uma cascata de 20 estágios.

Na Figura 6.1 (c) é dado um gráfico ilustrando o classificador do experimento 5 com 80% para treinamento e 20% para testes em uma cascata com 20 estágios. A área sobre a curva ROC tende a seguir uma linha próxima à área central do gráfico, porém com algumas atenuações, desta forma pode ser considerado como um classificador bom, porém não ideal.

O classificador apresentado na Figura 6.1 (d) pertence ao experimento 5 na proporção de 94% para treinamento e 6% para testes em um classificador de 15 estágios. A taxa de acerto desse classificador é de 82%, porém nas curvas ROC ele apresenta uma curva pequena e linear.

A curva ROC apresentada na Figura 6.1 (e), apresenta um bom resultado, haja visto que a taxa de verdadeiro positivo é maior e taxa de falsos positivos menor. Ela é resultado do classificador com 94% treinamento e 6% testes em um classificador de 20 estágios. De acordo com sua taxa de acerto e a curva ROC este é considerado o melhor classificador deste projeto. Com essa curva é possível perceber que existe uma boa diferenciação entre as classes, o que torna um sistema de detecção mais confiável.

Capítulo 7

Conclusões e trabalhos futuros

Este trabalho apresentou uma proposta de baixo custo para diagnóstico automático de malária em dispositivos móveis. Um modelo chamado *malariasystem* foi desenvolvido e implementado via linguagem de programação para plataformas móveis.

A principal contribuição do Malariasystem é o baixo custo e a possibilidade de ser utilizado por agentes de saúde pública em regiões endêmicas remotas, onde os recursos são escassos. Além disso os resultados podem ser avaliados in loco, facilitando o tratamento imediato.

Portanto, a partir do modelo proposto e dos resultados obtidos, foi possível avaliar a viabilidade da solução para plataformas móveis, destacando os seguintes aspectos:

- Aquisição da imagem: Foi possível avaliar a captura de imagens em um microscópio bacteriológico simples e um sistema de acoplamento experimental de baixo custo, ambos independentes de fabricantes, ou seja, é necessário somente ter os pré-requisitos propostos no Capítulo 5 para obter esse tipo de imagens, condicionados as lâminas contendo objetos sanguíneos. Além disto, com a captura de imagens é possível viabilizar essas imagens a um público geral, favorecendo o treinamento e a outras pesquisas em áreas relacionadas. Diversos equipamentos de capturas foram avaliados, e adaptados para ser um processo de baixo custo operacional.
- Pré-processamento da imagem: Através de algoritmos e filtros(vistos no Capítulo 5) foi possível realizar modificações nas imagens de forma a restringir o espaço de busca, conseqüentemente melhorar o desempenho da cascata de classificadores e diminuir as

taxas de falsos positivos, uma vez que o algoritmo passa a observar apenas dentro das células sanguíneas.

7.1 Contribuições

Portanto, a partir do modelo proposto e dos resultados obtidos foi possível avaliar a viabilidade da solução para plataformas móveis, desta forma é possível responder as perguntas introduzidas na Seção 1.3:

- (i) **É possível efetuar uma etapa de segmentação de imagens de modo a extrair informações de objetos dentro das células sanguíneas?**

A partir das imagens obtidas na Figura 5.1 é possível avaliar a qualidade da segmentação, de tal forma a restringir o espaço de buscas para a próxima etapa do algoritmo de reconhecimento. No trabalho de Anggraini [14] é obtido também um bom resultado de segmentação, caberia a trabalhos futuros avaliá-los.

- (ii) **De acordo com os fundamentos do *ShistoSystem*[15] é possível prover um bom classificador para diagnóstico para malária?**

O melhor classificador para este trabalho obteve uma taxa de acerto de 93%, com um conjunto de treinamento abaixo de 15% sugerido pela literatura [3], o que pode ser considerado um bom resultado considerando essas circunstâncias, haja disto que a detecção de parasitos de *plasmodium* não é uma tarefa trivial até para os profissionais mais treinados[18].

- (iii) **Em dispositivos móveis existe viabilidade para um sistema de baixo custo para diagnóstico de malária?**

Um sistema foi implementado (visto na Seção 5.6) para plataformas móveis resultando em dois aplicativos de detecção de parasitos do tipo *plasmodium*. O equipamento utilizado para testes foi um equipamento de baixo custo, porém não se limitando a este hardware em específico. Como o sistema operacional Android[9][8] é independente do hardware, o sistema pode ser avaliado em diversos dispositivos de fabricantes distintos. De certa forma é uma escolha estratégica uma vez que esse sistema operacional é o mais utilizado para plataformas móveis no mundo segundo a revista Forbes[47], portanto é possível uma avaliação e uso por diversas pessoas de todos os continentes.

7.2 Limitações

Embora a abordagem proposta tenha alcançado resultados preliminares algumas limitações são dadas ao modelo proposto:

- Este modelo é limitado a apenas uma fase do *plasmodium*, denominada trofozoíta em duas espécies *P.falciparum* e *P.Vivax*, conceituados no Capítulo 5.
- Não foi possível adquirir mais exemplos de imagens de *plasmodim*, devido à dificuldade de acesso aos laboratórios e centros de pesquisa, bem como não é uma tarefa trivial encontrar lâminas contendo *plasmodium*, pois mesmo quando elas existem é uma tarefa custosa do ponto de vista de tempo, em muitas vezes era possível adquirir apenas de 2 ou 3 imagens de parasitos em um dia de trabalho.
- É necessário obter um treinamento em diagnóstico de malária para poder efetuar a captura e marcação das imagens.

7.3 Trabalhos Futuros

Nesta Seção serão apresentadas sugestões para continuação deste trabalho:

- Efetuar avaliações do modelo proposto com outras bases de imagens.
- Testar os aplicativos desenvolvidos nessa dissertação em diversos laboratórios de pesquisa.
- Melhorar o método de pré-processamento proposto, de tal forma a segmentar a forma anelar do parasito na fase trofozoíta.
- Desenvolver um equipamento de microscopia portátil à ser acoplado ao Tablet, de tal forma a executar diagnóstico de malária em tempo real.
- Avaliar e comparar o algoritmo *adaboosting* utilizado nessa dissertação com outros algoritmos de reconhecimento de padrões como: Redes Neurais, K-vizinhos mais próximos, ou até mesmo criar uma solução híbrida.

-
- Desenvolver uma ferramenta de aprendizagem para estudantes de medicina em geral, com base neste trabalho.
 - Adquirir e treinar imagens contendo outras fases e variações do *plasmodium* nas quatro principais espécies : *P.falciparum*, *P.vivax*, *P.ovale* ,*P.malariae*.
 - Desenvolver um sistema de detecção de parasitos diferenciando as fases e espécies de malária.
 - Efetuar pré-processamento de imagem aliado a um sistema de correção de iluminação de imagens proposto por [48], e após treinar e avaliar os classificadores.
 - Efetuar uma avaliação do impacto epidemiológico da ferramenta, de forma a auxiliar nos sistemas de controle de epidemias.

7.4 Lista de Publicações

Com esta dissertação foram gerados os seguintes trabalhos:

1. Malaria System: a New Tool for Automatic Diagnosis of Malaria in Mobile Devices - has been **selected for presentation as Oral presentation** at Medicine 2.0: Social Media, Mobile Apps, and Internet/Web 2.0 in Health, Medicine and Biomedical Research at Torremolinos, Malaga ES[49].
2. A Proposal for Automatic Diagnosis of Malaria: Extended Abstract at the 22nd International World Wide Web Conference[16];
3. SchistoSystem - Inteligência Artificial para Diagnóstico Automático por Imagens[36];
4. SchistoSystem - Artificial Intelligence for Auto Diagnostic Imaging[35].

Outra(s) publicação(s) aceita(s) durante o período do mestrado:

1. Design de produtos de software dirigidos à inovação: um estudo de caso[50].

Referências Bibliográficas

- [1] C. for Disease Control and Prevotion, “Biology malaria,” tech. rep., Centers for Disease Control and Prevotion, 2013.
- [2] H. Pedrini and W. Schwartz, *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, 2008.
- [3] P. Viola and M. J. Jones, “Robust real-time face detection,” *Int. J. Comput. Vision*, vol. 57, pp. 137–154, May 2004.
- [4] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, 2008.
- [5] R. Szeliski, *Computer Vision: Algorithms and Applications*. Texts in Computer Science, Springer, 2010.
- [6] M. da Saúde, *Guia Prático de Tratamento da Malária no Brasil*. Brasília-DF: Ministério da Saúde, 2007.
- [7] M. Newman and F. Carranza, *Periodontia Clinica*. ELSEVIER (MEDICINA), 2007.
- [8] Android, “Android developer.” <http://developer.android.com>, 2014. Acessado em : 04/05/2014.
- [9] Z. Mednieks, L. Dornin, G. Meike, and M. Nakamura, *Programming Android*. Safari Books Online, O’Reilly Media, 2011.
- [10] L. N. O. B., A. Cesar, A. Jones, and B. Constanca, “The schisto track: A system for gathering and monitoring epidemiological surveys by connecting geographical information systems in real time,” *JMIR mhealth and uhealth*, vol. 2, mar 2014. <http://dx.doi.org/10.2196/mhealth.2859>.

-
- [11] S. HAYKIN, *Redes Neurais - 2ed.* BOOKMAN COMPANHIA ED, 2001.
- [12] S. Mavandadi, S. Dimitrov, S. Feng, F. Yu, U. Sikora, O. Yaglidere, S. Padmanabhan, K. Nielsen, and A. Ozcan, “Distributed medical image analysis and diagnosis through crowd-sourced games: A malaria case study,” *PLoS ONE*, vol. 7, 05 2012.
- [13] S. Kaewkamnerd, C. Uthaipibull, A. Intarapanich, M. Pannarut, S. Chaotheing, and S. Tongsimma, “An automatic device for detection and classification of malaria parasite species in thick blood film,” *BMC Bioinformatics*, vol. 13, no. Suppl 17, p. S18, 2012.
- [14] D. Anggraini, A. Nugroho, C. Pratama, I. Rozi, A. Iskandar, and R. Hartono, “Automated status identification of microscopic images obtained from malaria thin blood smears,” in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*, pp. 1–6, July 2011.
- [15] AC.Firmo, “Classificação adaboost com treinamento por enxame de partículas para diagnóstico da esquistossomose mansônica no litoral de pernambuco,” Master’s thesis, Universidade de Pernambuco, Recife, 2010.
- [16] A. D. Oliveira, G. Cabral, D. López, C. Firmo, F. Z. Serrat, and J. Albuquerque, “A proposal for automatic diagnosis of malaria: Extended abstract,” in *Proceedings of the 22Nd International Conference on World Wide Web Companion, WWW ’13 Companion*, (Republic and Canton of Geneva, Switzerland), pp. 681–682, International World Wide Web Conferences Steering Committee, 2013.
- [17] B. RB and B. L, *Doenças endêmicas : abordagens sociais, culturais e comportamentais [online]*. <http://books.scielo.org>: FIOCRUZ, 2000.
- [18] W. H. Organization, “World malaria report 2012,” tech. rep., World Health Organization, 2012.
- [19] W. Press, *Management of severe malaria: a practical handbook 3rd ed.* Geneva, Switzerland: World Health Organization, 2012.
- [20] V. Sortica, M. Cunha, M. Ohnishi, J. Souza, A. Ribeiro-dos Santos, N. Santos, S. Callegari-Jacques, S. Santos, and M. Hutz, “Il1b, il4r, il12rb1 and tnf gene polymorphisms are associated with plasmodium vivax malaria in brazil,” *Malaria Journal*, vol. 11, no. 1, p. 409, 2012.

-
- [21] S. de Vigilância em Saúde, *Manual de Diagnóstico Laboratorial da Malária*. Brasília-DF: Ministério da Saúde, 2005.
- [22] J. Silver, *Mosquito Ecology: Field Sampling Methods*. SpringerLink: Springer e-Books, Springer Science+Business Media B.V., 2008.
- [23] H. M. Gilles, *Tratamento da Malária Grave e Complicada*. Brasília-DF: Organização Mundial de Saúde, 1991.
- [24] F. das Chagas Oliveira Luiz, *Manual de Treinamento em Diagnóstico Laboratorial de Malária*. Brasília-DF: Ministério da Saúde, 2001.
- [25] W. Press, *Basic malaria microscopy*. Geneva, Switzerland: World Health Organization, 1991.
- [26] R. Gonzalez and R. Woods, *Processamento de imagens digitais*. Edgard Blucher, 2000.
- [27] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson/Prentice Hall, 2008.
- [28] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification, Wiley, 2001.
- [29] C. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Computer Vision, 1998. Sixth International Conference on*, pp. 555–562, 1998.
- [30] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–900–I–903 vol.1, 2002.
- [31] R. E. S. Yoav Freund, “A short introduction to boosting,” in *Journal of Japanese Society for Artificial Intelligence*, pp. 771–780, 1999.
- [32] R. Laganière, *OpenCV 2 Computer Vision Application Programming Cookbook: Over 50 Recipes to Master this Library of Programming Functions for Real-time Computer Vision*. Packt Open Source Pub., 2011.
- [33] S. Nagabhushana, *Computer Vision and Image Processing*. New Age International, 2005.
- [34] N. OTSU, “A threshold selection method from gray-level histograms,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, pp. 62–66, Jan 1979.

- [35] AC.Firmo, O. A.D., J. VIEGAS J., ALBUQUERQUE, and B. C. S, “Schistosystem - artificial intelligence for auto diagnostic imaging,” in *Free Themes Abstracts Epidemiology, Sanitation, Control, Elimination of Shistosomiasis and Intermediary Hosts in 13^o International Symposium on Schistosomiasis*, 2012.
- [36] AC.Firmo, O. A.D., J. VIEGAS J., ALBUQUERQUE, and B. C. S, “Schistosystem - inteligência artificial para diagnóstico automático por imagens,” in *Atas da Conferência IADIS Ibero-Americana Computação Aplicada 2013*, pp. 87–94, 2013.
- [37] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, pp. I–900–I–903 vol.1, 2002.
- [38] IOS, “Apple ios.” <https://www.apple.com/br/ios>, 2014. Acessado em : 04/05/2014.
- [39] WPHONE, “Windows phone.” <http://www.windowsphone.com>, 2014. Acessado em : 05/05/2014.
- [40] S. Boraks, *Programação com Java: Uma Introdução Abrangente*. McGraw-Hill, 2013.
- [41] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer Publishing Company, Incorporated, 1st ed., 2008.
- [42] G. M. Araujo, “Algoritmo para reconhecimento de características faciais baseado em filtros de correlação,” Master’s thesis, Pós-graduação em Engenharia Elétrica da Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.
- [43] L. R. P. de Paula, “Segmentação de imagens spect/gated-spect do miocárdio e geração de um mapa polar,” Master’s thesis, Universidade de São Paulo, São Paulo, 2011.
- [44] C. for Disease Control and Prevention, “Public health image library phil.” <http://phil.cdc.gov>, Apr. 2014.
- [45] wellcomeimages, “Wellcome images.” <http://wellcomeimages.org>, 2014. Acessado em : 01/05/2014.
- [46] C. E. Metz, “Basic principles of {ROC} analysis,” *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283 – 298, 1978.

-
- [47] E. S. Forbes, “Google and android have won mobile world congress by default.” <http://www.forbes.com/sites/ewanspence/2014/02/26/google-and-android-have-won-mobile-world-congress-by-default>, 2014. Acessado em : 15/05/2014.
- [48] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *Image Processing, IEEE Transactions on*, vol. 19, pp. 1635–1650, June 2010.
- [49] A. D. de Oliveira, M. E. Arruda, J. G. Prat, D. L. Codina, F. Z. Serrat, and J. O. Albuquerque, “Malaria system: a new tool for automatic diagnosis of malaria in mobile devices,” in *7th Word Congress Medicine 2.0*, 2014.
- [50] R. T. A. OLIVEIRA, A. D. Oliveira, B. C. AZEVEDO, and R. A. C. SOUZA, “Design de produtos de software dirigidos à inovação: um estudo de caso,” *Revista Brasileira de Administração Científica*, vol. 4, pp. 107 – 119, jul 2013.