

UnB – UNIVERSIDADE DE BRASÍLIA

FGA – FACULDADE GAMA

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
BIOMÉDICA**

**IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM
ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES**

LEONARDO ALVES DA COSTA

ORIENTADOR: Dr. MARCELINO MONTEIRO DE ANDRADE

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA BIOMÉDICA

PUBLICAÇÃO: 34A/2015

BRASÍLIA/DF: JULHO – 2015

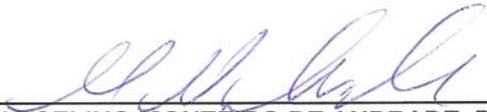
**UNIVERSIDADE DE BRASÍLIA
FACULDADE UNB GAMA
ENGENHARIA BIOMÉDICA**

**IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM
ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES.**

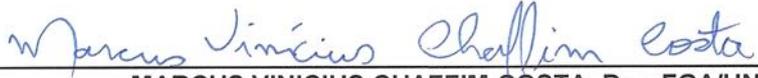
LEONARDO ALVES DA COSTA

DISSERTAÇÃO DE MESTRADO SUBMETIDA À FACULDADE UNB GAMA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA BIOMÉDICA.

APROVADA POR:



MARCELINO MONTEIRO DE ANDRADE, Dr. - FGA / UNB
(ORIENTADOR)



MARCUS VINICIUS CHAFFIM COSTA, Dr. - FGA/UNB
(EXAMINADOR INTERNO)



FLÁVIO DE BARROS VIDAL, Dr. – CIC / UNB
(EXAMINADOR EXTERNO)

BRASÍLIA, 03 DE JULHO DE 2015

BRASÍLIA/DF, 03 DE JULHO DE 2015.

FICHA CATALOGRÁFICA

LEONARDO ALVES DA COSTA	
IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES, [Distrito Federal] 2015.	
Nº. p. 90, 210 × 297 mm (FGA/UnB Gama, Mestre, Engenharia Biomédica, 2015).	
Dissertação de Mestrado – Universidade de Brasília. Faculdade UnB Gama. Programa de Pós-Graduação em Engenharia Biomédica.	
1. ANTIBIOGRAMA	2. MOMENTOS INVARIANTES
3. PROCESSAMENTO DIGITAL DE IMAGENS	
I. FGA UnB Gama / UnB.	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

DA COSTA, L. A. (2015). IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES. Dissertação de Mestrado em Engenharia Biomédica, Publicação 34A/2015. Programa de Pós-Graduação em Engenharia Biomédica, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, nº. p. 90.

CESSÃO DE DIREITOS

AUTOR: LEONARDO ALVES DA COSTA

TÍTULO: IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES

GRAU: Mestre

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização escrita do autor.

2015

Q 55/56 Lote 15/17 Bloco 2 Apartamento 638 Setor Central – Gama.
CEP 72.405-550 Brasília, DF – Brasil.

DEDICATÓRIA

Para minha esposa Sinéia Souza Santos Costa com amor e carinho.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por seu imenso amor. Ao meu orientador, o professor Dr. Marcelino Monteiro de Andrade, por sua imensa paciência, dedicação e persistência durante todo este trabalho, pois, sem ele, este trabalho não seria possível. Dedico, também, meus agradecimentos ao Edson Alves da Costa Júnior, pelo apoio técnico, e ao Thiago Espindola, pelo apoio emocional durante esta jornada.

RESUMO

IDENTIFICAÇÃO AUTOMÁTICA DE CARACTERES EM ANTIBIOGRAMAS COM USO DE MOMENTOS INVARIANTES

Autor: Leonardo Alves da Costa

Orientador: Prof. Dr. Marcelino Monteiro de Andrade

Programa de Pós-Graduação em Engenharia Biomédica

Brasília, 3 de Julho de 2015.

O antibiograma por disco-difusão é um procedimento laboratorial utilizado para determinar a suscetibilidade de micro-organismos em relação a um antibiótico. Neste exame, os antibióticos são impregnados em discos que são rotulados com caracteres alfanuméricos. A identificação automática destes rótulos é de crucial importância na automação deste antibiograma, visto que descrevem o antimicrobiano e sua respectiva concentração. Sendo assim, este trabalho apresenta uma técnica de descritores baseada em Momentos Invariantes com objetivo de reconhecer os padrões alfanuméricos presentes nos exames de antibiogramas por disco-difusão e sua aplicação no método Automático de Identificação do Antibiograma (AIA), que é parte integrante do projeto AUTOBAC. No experimento, foram utilizadas as técnicas de Momentos Invariantes em comparação ao uso do *Scale-invariant Feature Transform* (SIFT), técnica comercial utilizada pelo AIA. Para os Momentos Invariantes, a solução proposta apresentou 90,9% de identificações corretas e o SIFT apresentou 94,9% de identificações corretas, ou seja, a solução proposta apresentou um resultado apenas 4% menor do que a solução comercial proposta inicialmente pelo AIA, indicando a aplicabilidade da solução.

Palavras-chave: antibiograma. momentos invariantes. processamento digital de imagem.

ABSTRACT

AUTOMATIC CHARACTER IDENTIFICATION IN ANTIBIOGRAMS WITH INVARIANT MOMENTS

Author: Leonardo Alves da Costa

Supervisor: Prof. Dr. Marcelino Monteiro de Andrade

Post-Graduation Program in Biomedical Engineering

Brasília, July 3, 2015.

The antibiogram by disk diffusion is a laboratory procedure used to determine the relative susceptibility of microorganisms to an antibiotic. In this exam, antibiotics are impregnated discs labeled with alphanumeric characters. The automatic identification of these labels is crucial to antibiogram automation, as describing the antimicrobial and its respective concentration. Thus, this work presents a descriptors technique based on Invariant Moments in order to recognize the alphanumeric patterns present in antibiotic susceptibility tests by disk diffusion and its application in Automatic Identification Antibiogram (AIA) method, which is part of the AUTOBAC project. In the experiments, was achieved a comparison of the Invariant Moments techniques and the Scale-invariant Feature Transform (SIFT), a commercial technique used by the AIA. For the Invariant Moments, the proposed solution showed 90.9% of correct identifications and the SIFT with 94.9% of correct identifications, i.e., the proposed solution shows a result about only 4% lower than the commercial solution originally proposed by AIA, indicating the applicability of the solution.

Keywords: *antibiogram. invariant moments. digital image processing.*

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ACM	<i>Association for Computing Machinery.</i>
AIA	Algoritmo de Identificação Automática de Antibiograma.
CCD	<i>Charge Coupled Device.</i>
CLSI	<i>Clinical and Laboratory Standards Institute.</i>
DME	Diagnósticos Médicos Especializados.
FDA	<i>Food and Drug Administration.</i>
FINEP	Financiadora de Estudos e Projetos.
FIR	<i>Finite Impulse Response.</i>
IBICT	Instituto Brasileiro de Informação em Ciência e Tecnologia.
IEEE	<i>Institute of Electrical and Electronics Engineers.</i>
LIPIS	Laboratório de Instrumentação e Processamento de Imagens e Sinais.
LEI	Laboratório de Engenharia & Inovação.
MBC	<i>Minimum Bactericidal Concentration.</i>
MIC	<i>Minimum Inhibitory Concentration.</i>
OCR	<i>Optical Character Recognition.</i>
OMS	Organização Mundial de Saúde.
PDI	Processamento Digital de Imagens.
SIFT	<i>Scale-Invariant Feature Transform.</i>
UnB	Universidade de Brasília.
USP	Universidade de São Paulo.

LISTA DE TABELAS

3.1	Quantidade de rótulos separados por grupos considerando 12 rótulos distintos por grupo de controle.	56
4.1	Banco de Momentos Invariantes com o uso de 3 rótulos em sua formação (7 momentos invariantes).	61
4.2	Banco de Momentos Invariantes com o uso de 3 rótulos em sua formação (desvio padrão).	62
4.3	Banco de Momento Central com o uso de 3 rótulos em sua formação.	62
4.4	Momentos Invariantes de uma amostra aleatória do grupo experimental com 3 rótulos em sua formação.	62
4.5	Momentos Centrais de uma amostra aleatória.	63
4.6	Comparação entre os Momentos Invariantes com o uso da Distância Euclidiana ($L = 0,87$) em relação a Distância de Mahalanobis ($L = 50,5$).	64
4.7	Comparação entre os Momentos Invariantes com o uso da Distância de Mahalanobis ($L = 50,5$) em relação ao SIFT ($L = 10,5$).	64
4.8	Comparação entre os Momentos Invariantes com o uso da Distância de Mahalanobis ($L = 25$) em relação ao SIFT ($L = 10,5$).	65
4.9	Tempo de execução das técnicas aplicadas com o uso dos 156 rótulos do grupo experimental e 3 rótulos para o banco de momentos.	67
4.10	Tempo de execução das técnicas aplicadas com o uso dos 156 rótulos do grupo experimental com 3 rótulos para o banco de momentos e a utilização de banco de chaves SIFT.	67

LISTA DE FIGURAS

1.1	Exemplo típico de antibiograma por disco-difusão da DME.	21
2.1	Exemplo de pílula de antibiótico com um halo de inibição.	30
2.2	Fluxograma do Algoritmo de Identificação Automática.	43
3.1	Exemplo de placas utilizadas no banco do padrão Oxoid.	51
3.2	Fluxo de pré-processamento da imagem.	53
3.3	Fluxo de geração do banco de momentos invariantes.	54
3.4	Diagrama de aplicação dos Momentos Invariantes ao AIA.	55
3.5	Diagrama de execução do experimento.	57
4.1	Total de rótulos identificados por técnica de reconhecimento.	66
A.1	Exemplo de rótulos do padrão DME utilizados para o grupo de controle.	77

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Contextualização e formulação do problema	21
1.2	O projeto AUTOBAC	23
1.3	Objetivos	23
1.3.1	Objetivo geral	23
1.3.2	Objetivos específicos	24
1.4	Delimitação do estudo	24
1.5	Organização do trabalho	24
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Antibiogramas por disco-difusão	27
2.1.1	Indicações para a realização do teste de sensibilidade	27
2.1.2	Metodologia do antibiograma	28
2.1.3	Antibiograma quantitativo	28
2.1.4	Antibiograma qualitativo	29
2.1.5	Teste de diluição em Ágar	30
2.1.6	Meio de cultura	31
2.2	Técnicas de Reconhecimento Ótico de Caracteres	31
2.2.1	Técnica 1: Momentos Invariantes	33
2.2.2	Técnica 2: <i>Scale-Invariant Feature Transform</i>	35
2.3	Classificação e Correspondência de Características	39
2.3.1	Momentos Centrais	40
2.3.2	Distância Euclidiana	40
2.3.3	Distância de Mahalanobis	42
2.4	Ferramentas computacionais aplicadas	43
2.4.1	Algoritmo de Identificação Automática	43
2.4.2	Tesseract OCR	46
3	METODOLOGIA	49
3.1	Pesquisa Bibliográfica	49
3.2	Seleção dos materiais	51
3.3	Seleção de ferramentas	51

3.4	Proposta de Algoritmo de Identificação de Rótulos	52
3.4.1	Segmentação dos rótulos	52
3.4.2	Preparação das imagens	52
3.4.3	Criação do banco de momentos	53
3.4.4	Aplicação dos Momentos Invariantes ao AIA	54
3.5	Execução da simulação e dos experimentos	55
3.5.1	Geração do banco de momentos	57
3.5.2	Aplicação das medidas de comparação	58
3.5.3	Escolha do limiar de aceitação	58
4	RESULTADOS E DISCUSSÕES	61
5	CONSIDERAÇÕES FINAIS	69
5.1	Trabalhos futuros	70
	REFERÊNCIAS BIBLIOGRÁFICAS	71
	APÊNDICES	75
	A RÓTULOS UTILIZADOS NO GRUPO DE CONTROLE	77
	B SCRIPTS DO ALGORITMO DE IDENTIFICAÇÃO DE RÓTULOS	79

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E FORMULAÇÃO DO PROBLEMA

A análise da eficiência de antibióticos na resposta de um microrganismo sobre uma doença tem se tornado fundamental na área de medicina. Uma das técnicas de avaliação desta eficiência é realizada por meio do uso do teste *in vitro*, também chamada de antibiograma (Salgado *et al.*, 2001).

Assim, um antibiograma por disco-difusão corresponde a um procedimento de microbiologia realizado em laboratório, que possui o objetivo de determinar a suscetibilidade de micro-organismos (bactérias) com relação a antimicrobianos (antibióticos) (Costa *et al.*, 2012a). Os discos antimicrobianos são rotulados com caracteres alfanuméricos que representam o antibacteriano e sua respectiva concentração. Eles são regularmente identificados em procedimento visual por um especialista na prática laboratorial. A Figura 1.1 apresenta um exemplo de antibiograma por disco-difusão da empresa Diagnósticos Microbiológicos Especializados (DME).



Figura 1.1: Exemplo típico de antibiograma por disco-difusão da DME.

A partir de 1970, diversos laboratórios de microbiologia começaram a analisar de forma sistemática os dados de suscetibilidade, essencialmente de antibióticos betalactâmico (β -lactâmicos) e aminoglicosídeos. Este processo sistemático foi denominado como leitura interpretada do antibiograma (Courvalin, 1996).

Com o tempo, foi constatado que o antibiograma é um exame de baixo custo e de

resultados satisfatórios se forem seguidos os procedimentos padronizados pela CLSI, junto a uma estratégia de controle de qualidade. Mas existe, contudo, uma necessidade maior de automatização e agilidade no processo. Conforme Hadjigeorgiou *et al.* (2009), em alguns tratamentos, principalmente os tratamentos de infecções urinárias, o tempo gasto de 24 horas para o resultado do exame pode ser inadmissível. Este ponto abre possibilidades para novos projetos para a automação do processo ou para possibilidade de redução do tempo de incubação.

Devido a complexidade de interpretação dos dados de forma manual, referente a medição dos dados, é importante que hajam meios automáticos de medição. Contudo, a determinação e a medição automática do antibiograma ainda é um tópico aberto devido a vários problemas: os diferentes tipos de substratos cultivados, distorções nas imagens produzidos pelo rotulamento manual e a complexidade das texturas (Salgado *et al.*, 2001).

Em relação a essas e outras necessidades como as supracitadas, existem inúmeras técnicas de identificação de caracteres alfanuméricos, com diversos campos de aplicação, como sensoriamento remoto, transmissão e armazenamento de imagem médicas, radares, inspeção automática de componentes industriais, entre outros (Jain *et al.*, 2000; Flusser, 2000).

Considerando a identificação automática dos rótulos do disco-difusão de Costa *et al.* (2012a), foi aplicado o algoritmo de reconhecimento de padrões chamado de Algoritmo de Identificação Automática de Antibiograma (AIA), que utiliza a técnica *Scale-Invariant Feature Transform* (SIFT). Entretanto, essa técnica de identificação de padrões busca solução em cenários de maior complexidade, com relativa carga de processamento, natureza proprietária e restrições quanto a aplicação comercial (cobrança de *royalties* e/ou a restrição de uso por modelo de licenciamento de *software*). Por este motivo, iniciou-se uma busca por técnicas não proprietárias de identificação de padrões, que sejam mais adequadas ao problema da identificação dos rótulos em discos de antibiograma por disco-difusão. Os estudos levaram a uma solução baseada na aplicação da técnica clássica dos Momentos Invariantes.

Os Momentos Invariantes tornaram-se uma ferramenta para o reconhecimento de

objetos durante os últimos 40 anos (Hu, 1962; Flusser, 2000; Flusser e Suk, 2006). Sem dúvida, eles são os descritores de forma mais importantes e frequentemente mais utilizados (Flusser e Suk, 2006). Pois, mesmo sofrendo algumas limitações intrínsecas, eles servem como um método de referência para a avaliação do desempenho de outros descritores de forma. Apesar da grande quantidade de esforço e grande número de trabalhos publicados, ainda existem problemas abertos a serem resolvidos (Flusser e Suk, 2006).

Assim, este trabalho visa a substituição do uso da técnica SIFT (Costa *et al.*, 2012a) pela técnica de Momentos Invariantes, no sentido de garantir melhor desempenho computacional, ausência de restrições quanto a aplicação comercial pela equipe do AUTOBAC e simplificação do algoritmo de identificação de rótulos alfanuméricos em antibiograma por disco-difusão, sem alterar outras partes do processo atual do projeto.

1.2 O PROJETO AUTOBAC

Este trabalho é parte do projeto AUTOBAC, uma iniciativa FINEP (Edital – Finep 05/2010 – Produtos Médicos e Biomateriais), com o objetivo de desenvolver um sistema computacional embarcado para a realização automática da leitura interpretada do antibiograma, incluindo a elaboração de uma plataforma para a aquisição da imagem digital. O algoritmo desenvolvido para aquisição, processamento e interpretação de imagens digitais de antibiograma vem sendo desenvolvido desde 2010. As descrições e resultados já obtidos foram publicados em três congressos nacionais, indicando um método automático para identificação dos rótulos alfanuméricos dos antibióticos (Costa *et al.*, 2012a), da região de inibição (Costa *et al.*, 2012b) e do reconhecimento de caracteres nos rótulos (Costa *et al.*, 2014).

1.3 OBJETIVOS

1.3.1 Objetivo geral

Desenvolver e validar uma solução computacional automática para identificação de caracteres alfanuméricos, aplicada ao reconhecimento dos rótulos de antibiogramas por disco-difusão.

1.3.2 Objetivos específicos

1. Automatizar o processo de reconhecimento de antibacterianos em disco-difusão com técnica de processamento de imagem invariante ao deslocamento, rotação e translação;
2. Conceber uma solução computacional para ao reconhecimento dos rótulos de antibiograma por disco-difusão;
3. Auxiliar na identificação dos exames de antibiogramas por disco-difusão por meio do reconhecimento automáticos dos caracteres de seus rótulos;
4. Aplicar técnicas de reconhecimento ótico de caracteres (OCR – *Optical Character Recognition*);

1.4 DELIMITAÇÃO DO ESTUDO

O presente trabalho encontra-se inserido no escopo do projeto AUTOBAC, especificamente, na busca da implementação de soluções computacionais do reconhecimento de caracteres. Atualmente, o algoritmo de referência para reconhecimento de elementos alfanuméricos do projeto AUTOBAC utiliza, com resultados satisfatórios, o *Scale-Invariant Feature Transform* (Costa *et al.*, 2012a). Porém, a licença de uso da solução adotada é restrita à aplicação acadêmica, o que motivou a necessidade do desenvolvimento de uma outra solução, própria ao projeto, considerando o interesse comercial futuro da solução AUTOBAC. Além disso, é esperado que a solução do presente projeto apresente melhor desempenho computacional, considerando que a solução *Scale-Invariant Feature Transform* (SIFT) possui natureza não dedicada ao reconhecimento de caracteres alfanuméricos.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em 5 capítulos, incluindo este.

No Capítulo 2, é apresentada uma visão geral do referencial teórico, que visa a compreensão das tecnologias, conceitos, e padrões utilizados na indústria na identificação dos testes de disco-difusão. Logo, são abordados os seguintes temas: os antibiogramas

por disco-difusão, as técnicas de Reconhecimento Ótico de Caracteres (Momentos Invariantes e *Scale-Invariant Feature Transform*), a Classificação e Correspondência de Características e as ferramentas computacionais aplicadas neste estudo (o Algoritmo de Identificação Automática dos Rótulos e a ferramenta Tesseract OCR).

O Capítulo 3 detalha a metodologia utilizada no estudo e o Capítulo 4 descreve os resultados obtidos, uma análise dos seus significados e a sua aplicação ao projeto AUTOBAC.

O Capítulo 5 discute os pontos de maior importância que envolvem o tema deste estudo e as considerações finais do trabalho, assim como os trabalhos futuros que podem ser desenvolvidos a partir das propostas apresentadas neste documento.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ANTIBIOGRAMAS POR DISCO-DIFUSÃO

Os testes de sensibilidade aos agentes antimicrobianos referem-se às várias técnicas ou métodos laboratoriais *in vitro* utilizadas para determinar a sensibilidade e a resistência de determinado micro-organismo ao antimicrobiano (Amato Neto *et al.*, 2007; Tavares, 2006). Assim, este teste é utilizado para a verificação do efeito bactericida ou bacteriostático, bem como para a demonstração das drogas que exercem estes efeitos. O antibiograma consiste no cultivo do micro-organismo, cuja sensibilidade se quer avaliar, em presença de um ou vários antibióticos, verificando-se a ausência de desenvolvimento do micro-organismo no meio onde estão presentes as drogas ativas (Tavares, 2006).

Em bases ideais ou de experimentação científica, todos os micro-organismos bacterianos podem ser avaliados perante todos os antibióticos existentes. Na prática médica, contudo, o antibiograma está indicado para a averiguação da sensibilidade de bactérias e fungos que apresentam variabilidade na sensibilidade/resistência. O exame não é indicado para bactérias cuja sensibilidade aos antimicrobianos é constante, havendo pequena possibilidade de ocorrer resistência às drogas tradicionalmente ativas contra elas. Nestes casos, este é o antibiótico de escolha para o tratamento das infecções por elas causadas, o que torna a realização de antibiograma desnecessária (Amato Neto *et al.*, 2007; Tavares, 2006).

2.1.1 Indicações para a realização do teste de sensibilidade

Os testes de sensibilidade de determinado micro-organismo a agentes antimicrobianos são usados para orientar a seleção do agente mais adequado ao tratamento da infecção por ela causada, quando – mesmo sendo conhecida sua identidade – sua sensibilidade ou resistência não possa ser definida com segurança a partir dos dados disponíveis (Amato Neto *et al.*, 2007).

A escolha do método mais adequado para o teste de sensibilidade a antimicrobianos

depende do equipamento de que dispõe o laboratório, do número e dos tipos de micro-organismos e antimicrobianos a ser estudados e da finalidade a que se propõe a realização do teste (Amato Neto *et al.*, 2007).

O método mais amplamente utilizado é o teste da difusão em Ágar, embora muitos laboratórios clínicos já empreguem o teste de diluição (Amato Neto *et al.*, 2007).

2.1.2 Metodologia do antibiograma

O antibiograma pode ser quantitativo (método de diluição) ou qualitativo (método de difusão). O primeiro é o mais utilizado na prática, no qual uma concentração fixa do antibiótico é aplicada ao meio em que o micro-organismo está sendo cultivado (Tavares, 2006).

Modernamente, soluções comerciais automatizadas (BacT/Alert, Biogram, Bio Argos, Vitek, entre outros) vem sendo implantadas na rotina de laboratórios clínicos, permitindo a rápida realização de culturas e determinação da sensibilidade dos micro-organismos isolados de material clínico. Os sistemas automatizados utilizam-se de metodologia própria (tecnologia de densidade ótica, de fluorescência ou de radiometria, por exemplo) e permitem a identificação dos micro-organismos e sua sensibilidade aos antimicrobianos em prazos tão curtos quanto 3 ou 6 horas (Tavares, 2006).

Contudo, apesar de suas vantagens, deve-se tomar cuidado com o uso dos métodos automatizados, pois eles nem sempre representam a real sensibilidade dos micro-organismos identificados, o que pode comprometer a demonstração da resistência da droga (Salgado *et al.*, 2001; Amato Neto *et al.*, 2007; Tavares, 2006), dado que eles ainda são imprecisos em alguns aspectos e precisam ser melhorados (Costa *et al.*, 2012a).

2.1.3 Antibiograma quantitativo

O método de diluição é realizado em meios líquidos ou sólidos, aplicando-se concentrações crescentes do antibiótico aos meios de cultivo de bactérias, determinando-se, dessa maneira, a menor concentração capaz de inibir seu crescimento, isto é, a concentração inibitória mínima (MIC – *Minimum Inhibitory Concentration*), correspondente à concentração bacteriostática. Por esse método pode-se determinar, também, a concentração

bactericida mínima do medicamento (MBC – *Minimum Bactericidal Concentration*). O método de diluição é o habitualmente empregado para a verificação da sensibilidade dos micro-organismos anaeróbios aos antibióticos ou em determinadas situações clínicas. Entretanto, poucos laboratórios de rotina estão aparelhados para realizá-lo, devido à dificuldades técnicas e ao elevado custo do teste (Tavares, 2006).

Atualmente, o método de difusão em disco, similar ao usado para os micro-organismos aeróbios, foi padronizado para o antibiograma dos anaeróbios, o que facilitou a execução de rotina do exame (Tavares, 2006).

2.1.4 Antibiograma qualitativo

O antibiograma de difusão pelo método de disco é o mais utilizado, não só por sua facilidade, mas também pela rapidez do resultado obtido. A fim de se obter um resultado ainda mais rápido, o teste pode ser realizado antes mesmo da completa identificação do micro-organismo, bastando, para isso, que os discos sejam aplicados ao meio de cultura no momento da semeadura inicial do material. Em 24 horas, observam-se quais as drogas que inibiram o seu crescimento, tendo-se assim o resultado esperado antes mesmo que o micro-organismo tenha sido devidamente identificado (Tavares, 2006).

O resultado do antibiograma é dado pelo diâmetro do halo de inibição, medido em milímetros e padronizado de acordo com a droga, com a concentração utilizada e com o micro-organismo. Em geral, o resultado é expresso com os termos “sensível”, “moderadamente resistente” e “resistente”. No antibiograma realizado adequadamente, de acordo com as normas e padrões estabelecidos pela Organização Mundial de Saúde (OMS) e pela *Food and Drug Administration* (FDA), dos Estados Unidos, há correlação entre o diâmetro do halo de inibição e a concentração inibitória mínima (Amato Neto *et al.*, 2007; Tavares, 2006). Em alguns novos antibióticos betalactâmicos também existe o termo “moderadamente sensível” e permitem identificar cepas que possam requerer doses maiores que as habituais para o êxito do tratamento (Amato Neto *et al.*, 2007). A Figura 2.1 apresenta um exemplo de pílula de antibiótico com um halo de inibição (Costa *et al.*, 2012a).

É necessário lembrar, por outro lado, que a resistência *in vitro* pode não corresponder à resposta terapêutica, por vezes favorável com o uso da droga, devido às elevadas

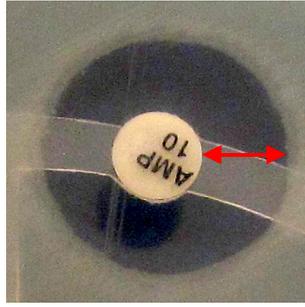


Figura 2.1: Exemplo de pílula de antibiótico com um halo de inibição.

concentrações que o antibiótico alcança em determinados setores do organismo. Assim, um micro-organismo resistente *in vitro* pode ser sensível *in vivo* a concentrações altas da droga, alcançadas, por exemplo, nas vias urinárias ou biliares. Eis o motivo por que o termo resistência é relativo e relacionado sempre às concentrações sanguíneas (Tavares, 2006).

Por fim, deve-se ressaltar que não há necessidade de se testar a ação de todas as drogas em um antibiograma, visto que existem grupos de antibióticos que apresentam a mesma atividade (Tavares, 2006) e, apesar de ressalvas, o teste qualitativo constitui procedimento efetivo para a maioria das exigências da rotina do laboratório clínico, podendo ser complementado por testes de diluição quando resultados quantitativos forem necessários (Amato Neto *et al.*, 2007).

2.1.5 Teste de diluição em Ágar

O teste de diluição em Ágar é realizado adicionando-se ao meio com diluições seriadas dos agentes antimicrobianos, momentos antes de distribuí-las em placas de Petri. Este teste apresenta três vantagens em relação ao teste de diluição em caldo (Amato Neto *et al.*, 2007; Tavares, 2006):

- a) A possibilidade de testar um grande número de cepas simultaneamente;
- b) A possibilidade de detectar heterogeneidade microbiana ou contaminação;
- c) A possibilidade de adicionar sangue total ou componentes do sangue ao meio, permitindo, assim, a realização do teste com alguns micro-organismos exigentes do ponto de vista nutritivo, que não poderiam ser testados satisfatoriamente em meio de cultura líquido.

O meio de cultura para o teste de diluição em Ágar é preparado e resfriado entre 45° a 50 °C; em seguida, soluções de antimicrobianos diluídos são adicionadas na proporção de uma parte de antimicrobiano para nove partes do meio. O meio é, então, distribuído em número adequado de placas de Petri. Se necessário, sangue ou algum de seus componentes podem ser adicionados depois do antimicrobiano. Para cada série de diluições, uma placa com o meio de Müller-Hinton não contendo antimicrobiano é preparada como controle (Bauer *et al.*, 1966). A suspensão padronizada do micro-organismo é, então, semeada, assumindo aspecto de mancha que ocupa área correspondente a um círculo com 5 a 8 mm de diâmetro. Pode ser aplicada com alça calibrada para 0,001 ml e não deve ser espalhada. Deixa-se secar e, na sequência, várias cepas poderão ser testadas na mesma placa. As placas são incubadas a 35 °C durante 16 a 20 horas antes da leitura. Verifica-se previamente a placa controle, sem antimicrobiano, para comprovar que cada cepa testada foi capaz de crescer adequadamente. As placas restantes são, então, examinadas para se determinar a concentração inibitória mínima do crescimento (Amato Neto *et al.*, 2007).

2.1.6 Meio de cultura

Como ainda não existe meio de cultura ideal, na atualidade, o habitualmente recomendado é o meio de Müller-Hinton, cujo pH deve estar entre 7,2 e 7,4 à temperatura ambiente. Para o crescimento de algumas bactérias exigentes, acrescenta-se a esse meio 5% de sangue desfibrinado de carneiro, de cavalo ou de outros animais, isentos de antibióticos. Assim, deve-se distribuir um volume de 60 a 70 ml do meio em placas de 150 mm e 25 a 30 ml em placas de 100 mm. Essas placas podem ser conservadas em geladeira (entre 2° a 8 °C) durante 7 dias, se embaladas em sacos plásticos fechados para evitar-se a dessecação. Não se deve permitir a presença de gotas de água na superfície das placas, sendo que se colocadas na estufa durante 10 a 20 min, o excesso de umidade evapora (Amato Neto *et al.*, 2007).

2.2 TÉCNICAS DE RECONHECIMENTO ÓTICO DE CARACTERES

O Reconhecimento Ótico de Caracteres (OCR – *Optical Character Recognition*) faz parte de uma família de técnicas de reconhecimento automático, como o reconhecimento de fala e de rádio frequência (Mori *et al.*, 1999), e é utilizada para o reconhecimento

automático de caracteres por meio de um mecanismo ótico (Mithe *et al.*, 2013).

O OCR pode reconhecer textos escritos à mão, assim como os textos impressos (Mithe *et al.*, 2013). Contudo, o seu desempenho está associado diretamente a qualidade do documento de entrada (Mithe *et al.*, 2013). Ele foi desenvolvido para processar imagens que consistem em quase sua totalidade de textos e com poucos itens (Mithe *et al.*, 2013).

Dentre as diversas técnicas existentes, pode-se citar (Mithe *et al.*, 2013):

- **Distribuição de pontos:** abrange técnicas que extraem recursos com base na distribuição estatística dos pontos. Estas características são geralmente tolerantes a distorções e variações de estilo. Algumas das técnicas típicas dentro desta área são:

- **Zoneamento:** o retângulo que circunscreve o caractere é dividido em várias sobreposições, ou não-sobreposições, regiões e densidades de pontos pretos e são calculados e utilizados como recursos.
- **Momentos:** as características são momentos de pontos sobre um centro escolhido, por exemplo, o centro de gravidade, ou um sistema de coordenadas escolhido.
- **Cruzamentos e distâncias:** as características são encontrados a partir do número de vezes que a forma de caracteres é atravessada por vetores ao longo de certas direções. Esta técnica é frequentemente utilizada por sistemas comerciais, porque pode ser realizado a alta velocidade e requer pouca complexidade.

A forma dos caracteres são medidas ao usar a técnica de distância determinados comprimentos ao longo dos vetores que os atravessam. Por exemplo, o comprimento dos vetores de dentro do limite do caractere.

- **N-tuplas:** a ocorrência conjunta relativa de pontos pretos e brancos (primeiro plano e fundo) em determinadas ordens são utilizadas como recursos.
- **Características locais (*Loci*):** são gerados vetores verticais e horizontais para cada ponto no fundo do caractere. O número de vezes que os segmentos

de linha que descrevem os caracteres são atravessados por estes vetores são usados como características.

- **Transformações e expansões de séries:** estas técnicas ajudam a reduzir a dimensionalidade do vetor de características e as características extraídas podem ser feitas invariante a deformações globais, como a translação e rotação. As transformações utilizados podem ser Fourier, Walsh, Haar, Hadamard, Karhunen-Loeve, Hough, Transformação do Eixo Principal, entre outras.

Muitas destas transformações são baseados na curva que descreve o contorno dos caracteres. Isto significa que estas características são muito sensíveis ao ruído que afetam o contorno do caractere, como lacunas não intencionais no contorno (Mithe *et al.*, 2013).

- **Análise estrutural:** as características geométricas e topologias estruturais de um símbolo são extraídos durante a análise estrutural. Por meio dessas características é que se tenta descrever a formação do caractere e alguns dos recursos mais usados são traçados. Em comparação com outras técnicas, a análise estrutural dá características com alta tolerância a variações de ruído e de estilo. No entanto, as características são apenas moderadamente tolerantes à rotação e translação. Infelizmente, a extração destes elementos não é trivial e, de alguma maneira, ainda é uma área em pesquisa (Mithe *et al.*, 2013).

Dentre os tipos supracitados, vale enfatizar duas técnicas de distribuição de pontos: os Momentos Invariantes e o *Scale-Invariant Feature Transform*, assim como a Classificação e Correspondência de Características associada a elas. Serão citadas, também, as duas ferramentas computacionais aplicadas durante a solução proposta neste trabalho.

2.2.1 Técnica 1: Momentos Invariantes

Um momento bidimensional de ordem $(p+q)$ de uma função de distribuição de densidade $\rho(x, y)$ pode ser definida nos termos das integrais de Riemann como (Hu, 1962):

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x, y) dx dy \quad (2.1)$$

com $p, q = 0, 1, 2, \dots$

No caso discreto (Gonzalez e Woods, 2002),

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (2.2)$$

onde $I(x, y)$ é a intensidade de tons de cinza da imagem nas coordenadas (x, y) .

Hu (1962) definiu os momentos centrais como:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (2.3)$$

e no caso discreto,

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (2.4)$$

onde

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}} \quad (2.5)$$

Desta forma, os momentos centrais até a ordem 3 podem ser simplificados conforme (Hu, 1962):

$$\mu_{00} = M_{00} \quad (2.6)$$

$$\mu_{01} = \mu_{10} = 0 \quad (2.7)$$

$$\mu_{11} = M_{11} - \bar{x}M_{01} = M_{11} - \bar{y}M_{10} \quad (2.8)$$

$$\mu_{20} = M_{20} - \bar{x}M_{10} \quad (2.9)$$

$$\mu_{02} = M_{02} - \bar{y}M_{01} \quad (2.10)$$

$$\mu_{21} = M_{21} - 2\bar{x}M_{11} - \bar{y}M_{20} + 2\bar{x}^2M_{01} \quad (2.11)$$

$$\mu_{12} = M_{12} - 2\bar{y}M_{11} - \bar{x}M_{02} + 2\bar{y}^2M_{10} \quad (2.12)$$

$$\mu_{30} = M_{30} - 3\bar{x}M_{20} + 2\bar{x}^2M_{10} \quad (2.13)$$

$$\mu_{03} = M_{03} - 3\bar{y}M_{02} + 2\bar{y}^2M_{01} \quad (2.14)$$

Para os momentos invariantes de escala, os momentos devem ser definidos como η_{ij} , com $i + j \geq 2$, divididos pelo momento μ_{00} propriamente escalonado, conforme mostra

a Equação 2.15 (Flusser, 2000).

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{1+(i+j)/2}} \quad (2.15)$$

Hu (1962) definiu sete momentos invariantes para a rotação de objetos 2D como (Gonzalez e Woods, 2002):

$$\phi_1 = \mu_{20} + \mu_{02} \quad (2.16)$$

$$\phi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \quad (2.17)$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \quad (2.18)$$

$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \quad (2.19)$$

$$\begin{aligned} \phi_5 = & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) \times [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03}) \times [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (2.20)$$

$$\phi_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) \quad (2.21)$$

$$\begin{aligned} \phi_7 = & (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12}) \times [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03}) \times [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \end{aligned} \quad (2.22)$$

O momento central do objeto $f(x, y)$ e (\bar{x}, \bar{y}) são as coordenadas de centroide do objeto. Hu (1962) também apresentou como computar a invariância de escala e demonstrou o poder discriminatório destas funções no caso de reconhecimento de caracteres impressos (Flusser e Suk, 2006; Flusser, 2000). Para que os momentos sejam invariantes de translação, escala e rotação, os Momentos Invariantes de rotação devem utilizar o momento escalonado e rotacionado, conforme definido em η_{ij} (Equação 2.15).

Em sua conclusão, Hu (1962) definiu que dentre os 7 momentos, os primeiros 6 são invariantes de translação, escala e rotação. Contudo, o sétimo é invariante a torção, o que ajuda a distinguir imagens espelhadas.

2.2.2 Técnica 2: *Scale-Invariant Feature Transform*

O *Scale-Invariant Feature Transform* (SIFT) é um algoritmo da Visão Computacional utilizado para detectar e descrever características (*features*) em imagens. Ele foi proposto

por David Lowe em 1999 (Lowe, 1999) e, posteriormente, patentado nos EUA em 2004, em posse da University of British Columbia (Lowe, 2004). Ele pode ser dividido na identificação, na estabilização, na indexação e na correspondência das chaves SIFT. Será apresentado também a solução utilizada para os parâmetros afins.

Identificação das chaves SIFT

Para a identificação das chaves, é necessário identificar locais na escala de espaço da imagem que são invariantes em relação a translação, escala e rotação da imagem e que sejam minimamente afetadas pelo ruído e pequenas distorções (Lowe, 1999).

Para alcançar a invariância a rotação e um nível elevado de eficiência, são selecionadas as chave locais nos máximos e mínimos de uma diferença da função de Gauss aplicada em escala de espaço. Isto pode ser calculado de forma muito eficiente por criar uma imagem pirâmide com reamostragem entre cada nível (Lowe, 1999). Além disso, localiza-se os pontos-chave em regiões e escalas de alta variação, tornando esses locais particularmente estáveis para caracterizar a imagem (Lowe, 1999).

Como a função Gaussiana 2D é separável, a sua convolução com a imagem de entrada pode ser eficientemente computada pela aplicação de duas passadas da função Gaussiana 1D nas direções horizontais e verticais conforme a Equação 2.23 (Lowe, 1999).

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \quad (2.23)$$

Para localização das chaves, todas as operações de suavização são feitas usando $\sigma = \sqrt{2}$, que pode ser aproximada com precisão suficiente utilizando um núcleo 1D com 7 pontos de amostra (Lowe, 1999).

A imagem de entrada é primeiro envolvida com a função gaussiana usando $\sigma = \sqrt{2}$, chamada de imagem A. Em seguida, é efetuada uma segunda suavização incremental de $\sigma = \sqrt{2}$, chamada de imagem B, que tem agora uma suavização efetiva de $\sigma = 2$. A diferença da função de Gauss é obtida pela subtração da imagem B de A, o que resulta em uma proporção de $2/\sqrt{2} = \sqrt{2}$ entre as duas Gaussianas (Lowe, 1999).

Para gerar o próximo nível pirâmide, é realizada uma reamostragem da imagem B

já suavizada utilizando uma interpolação bilinear com um pixel espaçamento de 1,5 em cada direção (Lowe, 1999). O espaçamento de 1,5 significa que cada nova amostra será uma combinação linear constante de quatro pixels adjacentes. Isto é eficiente para computar e minimizar serrilhamentos (*aliasing*) que decorreriam da alteração dos coeficientes de reamostragem (Lowe, 1999).

Estabilização das chaves SIFT

Para caracterizar a imagem em cada chave local, a imagem suavizada em cada nível da pirâmide é processada para extrair gradientes e orientações de imagem (Lowe, 1999). Em cada pixel, A_{ij} , a magnitude do gradiente de imagem, M_{ij} , e orientação, R_{ij} , são calculados utilizando as diferenças de pixels conforme (Lowe, 1999):

$$M_{ij} = \sqrt{(A_{ij} - A_{i+1,j})^2 + (A_{ij} - A_{i,j+1})^2} \quad (2.24)$$

$$R_{ij} = \text{atan2}(A_{ij} - A_{i+1,j}, A_{i,j+1} - A_{ij}) \quad (2.25)$$

onde a função atan2 pode ser definida como o arco tangente de dois parâmetros conforme apresentado na Equação 2.26 (Organick, 1966, p. 42).

$$\text{atan2}(y, x) = \begin{cases} 2 \arctan\left(\frac{y}{\sqrt{x^2+y^2}+x}\right), & \text{se } x > 0 \text{ ou } y \neq 0. \\ \pi, & \text{se } x < 0 \text{ e } y = 0. \\ \text{indefinido}, & \text{se } x = 0 \text{ e } y = 0. \end{cases} \quad (2.26)$$

Cada chave local é atribuída uma orientação canônica para que os descritores de imagem sejam invariantes à rotação (Lowe, 1999). A fim de torná-la tão estável quanto possível contra mudanças de iluminação ou de contraste, a orientação é determinada pelo pico do histograma das orientações das imagens gradientes locais (Lowe, 1999). A orientação do histograma é criada usando uma janela Gaussiana ponderada com σ de 3 vezes maior do que a escala atual de suavização. Estas ponderações são multiplicadas pelos valores de gradiente limiarizadas e acumuladas no histograma em locais correspondentes à orientação, R_{ij} (Lowe, 1999).

Indexação e correspondência

Para indexação, é necessário armazenar as chaves SIFT para imagens de exemplo e, em seguida, identificar as chaves correspondentes de novas imagens (Lowe, 1999). O problema de identificar as chaves mais semelhantes em vetores dimensionais elevados é conhecida por ter alta complexidade se é necessária uma solução exata (Lowe, 1999).

Solução para os parâmetros afins

A tabela *hash* é pesquisada para identificar todos os grupos de pelo menos 3 entradas em um *bin* e os *bins* são classificados em ordem decrescente de tamanho (Lowe, 1999). Cada um destes grupos são, então, sujeitos a um procedimento de verificação em que uma solução de mínimos quadrados é efetuada para os parâmetros de projeção afim relativas ao modelo para a imagem (Lowe, 1999).

A transformação afim do ponto do modelo $[x \ y]^T$ para um ponto na imagem $[u \ v]^T$ pode ser escrita como (Lowe, 1999):

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2.27)$$

onde o a translação do modelo é $[t_x \ t_y]^T$ e a rotação afim, escala e alargamento são representadas pelos parâmetros m_i .

Assim, para resolver os parâmetros de transformação, a Equação 2.27 pode ser reescrita como (Lowe, 1999):

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & & & \\ & & \dots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix} \quad (2.28)$$

onde é apresentada apenas uma combinação única, mas qualquer número de novas combinações podem ser adicionadas com cada combinação contribuindo com mais duas

linhas para a primeira e última matriz. Pelo menos 3 combinações são necessários para fornecer uma solução.

De outra maneira, pode-se reescrever a solução conforme a equação linear (Lowe, 1999):

$$A\hat{x} \approx b \quad (2.29)$$

A solução de mínimos quadrados para os parâmetros x pode ser determinada por resolver as equações normais correspondentes (Lowe, 1999):

$$\hat{x} = (A^T A)^{-1} A^T b \quad (2.30)$$

o que minimiza a soma dos quadrados das distâncias das localizações do modelo projetados para as imagens locais correspondentes.

2.3 CLASSIFICAÇÃO E CORRESPONDÊNCIA DE CARACTERÍSTICAS

Para que as características possam compor uma correspondência entre os caracteres, elas precisam ser classificadas, o que é definido como o processo de identificação de cada característica e sua atribuição a uma classe de caracteres correta (Mithe *et al.*, 2013).

A correspondência de características abrange os grupos de técnicas com base em medidas de similaridade em que a distância entre o vector de características, que descreve o caractere extraído, e a descrição de cada característica é calculada (Mithe *et al.*, 2013). Podem ser usadas diferentes medidas de distância, mas o comum é utilizar a Distância Euclidiana (Mithe *et al.*, 2013). Este classificador de distância mínima funciona bem quando as classes são bem separadas, isto é, quando a distância entre os meios é grande em comparação com a expansão de cada classe (Mithe *et al.*, 2013).

Desta maneira, serão apresentada 3 maneiras de realizar estas classificações que se aplicam a solução proposta: o uso de Momentos Centrais com a Distância Euclidiana, conforme apresentada por Hu (1962); o uso dos Momentos Invariantes com a Distância

Euclidiana e o uso dos Momentos Invariantes com a Distância de Mahalanobis.

2.3.1 Momentos Centrais

Para o uso dos momentos centrais, Hu (1962) utilizou um ponto definido como:

$$X = \mu_{20} + \mu_{02} \quad (2.31)$$

$$Y = \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \quad (2.32)$$

onde μ_{20} , μ_{02} e μ_{11} são momentos normalizados com respeito ao tamanho e o ponto (X, Y) em um espaço bidimensional é utilizado pela representação do padrão.

Em seguida, Hu (1962) definiu que um grupo de padrões representados por (X_i, Y_i) com $i = 1, 2, \dots, n$ pode ter sua distância obtida dos pontos (X, Y) e (X_i, Y_i) :

$$d_i = \sqrt{(X - X_i)^2 + (Y - Y_i)^2} \quad (2.33)$$

considerando a distância d_{min} conforme:

$$d_{min} = \min_i d_i \quad (2.34)$$

de modo que é escolhida a distância d_k satisfazendo a condição $d_k = d_{min}$ (se mais de uma distância satisfizer esta condição, uma d_k é escolhida aleatoriamente). Então, d_k é comparado com um limiar L , se a condição $d_k \leq L$ for satisfeita, o padrão é reconhecido, senão ele é rejeitado (Hu, 1962). Para que este algoritmo tenha um desempenho melhor, Hu (1962) também citou que é importante treinar o sistema com a criação de um banco de momentos centrais e ajustar o limiar baseado nas amostras de treinamento.

2.3.2 Distância Euclidiana

A distância da linha reta entre dois pontos em um espaço N -dimensional é conhecido como a Distância Euclidiana. Essa distância entre dois vetores de características pode ser apresentada como (Cash e Hatamian, 1987):

$$D_E = \sqrt{\sum_{i=1}^N (F_{L_i} - F_{I_i})^2} \quad (2.35)$$

onde F_{L_i} é a i -ésima característica do banco de características e F_{I_i} é a i -ésima característica de entrada.

Contudo, Cash e Hatamian (1987) mostraram que esta distância é mais efetiva na comparação de características quando associadas ao uso de um fator de peso baseado no desvio padrão do banco de funcionalidades.

Desta forma, a Distância Euclidiana entre dois vetores de funcionalidade é definida como (Cash e Hatamian, 1987):

$$D_E = \sqrt{\sum_{k=1}^N w_k (F_{L_k} - F_{I_k})^2} \quad (2.36)$$

onde F_L é o banco de funcionalidades, F_I é a funcionalidade de entrada e w_k é o k -ésimo fator de peso.

Cash e Hatamian (1987) também definiu o cálculo do peso conforme a Equação 2.37:

$$w_k = \frac{1}{\bar{\sigma}_k} \quad (2.37)$$

onde o denominador $\bar{\sigma}_k$ é a média do desvio padrão para as k -ésimas funcionalidades em todos os rótulos. Essa média pode ser calculada conforme a Equação 2.38.

$$\bar{\sigma}_k = \frac{1}{n} \sum_{j=1}^n \left[\frac{m \sum_{i=1}^m F_{j_i}^2 - \left(\sum_{i=1}^m F_{j_i} \right)^2}{m(m-1)} \right]^{1/2} \quad (2.38)$$

onde n é o número de classes, m é o número de rótulos em cada classe e F_{j_i} é a característica para cada i -ésimo rótulo para a j -ésima classe.

No caso de uma distância de intraclasses grande, Cash e Hatamian (1987) também sugeriu que o peso pode ser obtido conforme a Equação 2.39:

$$w_k = \frac{\sigma_k}{\bar{\sigma}_k} \quad (2.39)$$

onde o fator $\bar{\sigma}_k$ é o mesmo fator do peso apresentado em Equação 2.38.

2.3.3 Distância de Mahalanobis

Criada por Mahalanobis (1936), ela define a distância ponderada entre os vetores de características de entrada e a média do banco de características para uma classe particular. O peso usado é recíproca da variância das características intraclasse (Cash e Hatamian, 1987).

A Distância de Mahalanobis é a observação de $x = (x_1, x_2, x_3, \dots, x_n)^T$ de um grupo de observações da média $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_n)^T$ e a matriz de covariância S definida conforme a Equação 2.40 (De Maesschalck *et al.*, 2000).

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (2.40)$$

Ela também pode ser definida como uma medida de dissimilaridade entre dois vetores aleatórios \vec{x} e \vec{y} de mesma distribuição e a matriz de covariância S (De Maesschalck *et al.*, 2000):

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (2.41)$$

Se a matriz de covariância é uma matriz identidade, esta distância pode ser reduzida a Distância Euclidiana (De Maesschalck *et al.*, 2000). Se a matriz de covariância é diagonal, então a medida de distância resultante é chamada de Distância Euclidiana Normalizada (De Maesschalck *et al.*, 2000) e pode ser definida como:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^N \frac{(x_i - y_i)^2}{s_i^2}} \quad (2.42)$$

onde s_i é o desvio padrão de x_i e y_i sobre um mesmo conjunto de amostras.

Desta maneira, Cash e Hatamian (1987) sugeriu seu uso na combinação de características conforme a Equação 2.43.

$$D_M = \sqrt{\sum_{i=1}^N \frac{(\bar{F}_{L_i} - F_{I_i})^2}{\sigma_{L_i}^2}} \quad (2.43)$$

onde \bar{F}_{L_i} é a média de características do banco, F_{I_i} é a característica de entrada e σ_{L_i}

é o desvio padrão de cada funcionalidade do banco de funcionalidades.

2.4 FERRAMENTAS COMPUTACIONAIS APLICADAS

Dentre as várias ferramentas de OCR existentes, foram escolhidos o Algoritmo de Identificação Automática (AIA), por ser integrante do projeto AUTOBAC, e a ferramenta *Open Source* Tesseract OCR, por ser uma das ferramentas disponíveis de forma livre que mais se aproxima das ferramentas comerciais (Mithe *et al.*, 2013).

2.4.1 Algoritmo de Identificação Automática

O algoritmo de segmentação dos rótulos segue conforme a proposta e desenvolvimento de Costa *et al.* (2012b), que foi desenvolvido no Matlab versão 7.6 e seu modelo esquemático é apresentado na Figura 2.2.

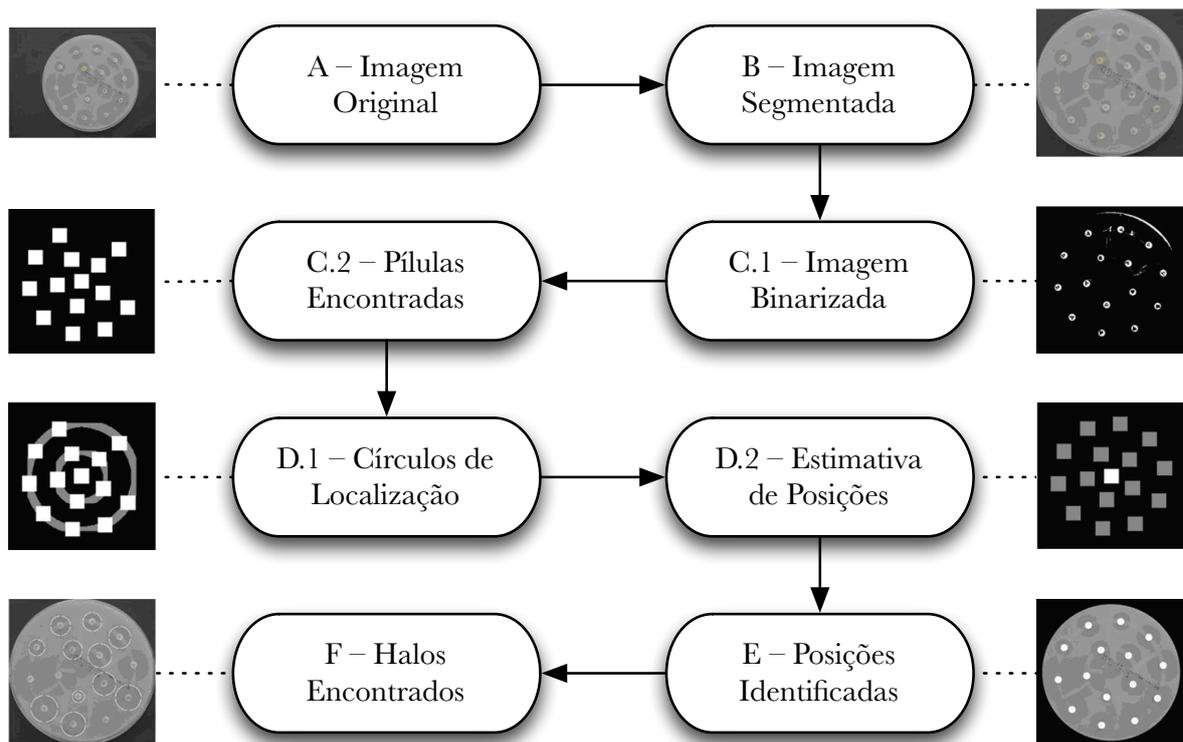


Figura 2.2: Fluxograma do Algoritmo de Identificação Automática (Costa *et al.*, 2012a).

Vale lembrar que o AUTOBAC em seu projeto atual visa a aquisição e construção de um equipamento de aquisições de imagens que possibilite a aquisição de imagens sem transformações afins.

O Algoritmo de Identificação Automática (AIA) pode ser dividido em três etapas

fundamentais: estimativa das posições dos discos na imagem segmentada, identificação das posições e identificação das regiões de inibição dos discos (halos). Estas etapas são precedidas por uma segmentação geral e um processamento morfológico, que preparam as imagens para as etapas principais.

Após a segmentação da região de interesse (Figura 2.2-B), é gerada uma imagem binária dos discos antimicrobianos através de uma binarização por histograma da imagem segmentada pelo método de Otsu (Otsu, 1979), utilizando um limiar de 97% da amplitude dos pixels (Figura 2.2-C.1).

Para isolar os discos dos demais elementos resultantes da binarização, um processamento morfológico é aplicado em cada região rotulada da imagem binária. Assim, foram analisadas a razão entre a altura (A), largura (L) e a área (A_r) das regiões, sendo considerados valores aceitáveis para A/L e D os intervalos $[0.9, 1.1]$ e A_r do círculo (πr^2) compreendido entre $\pi(2.7 \text{ mm})^2$ e $\pi(3.3 \text{ mm})^2$. Esses parâmetros permitiram eliminar todos os objetos que não fossem quase circulares, com aproximadamente 6 cm de diâmetro da imagem binária. A Figura 2.2-C.2 ilustra as regiões classificadas pelos parâmetros A/L e A_r .

Para estimar a posição dos discos não classificados, foram criados dois anéis semelhantes aos encontrados no polisensidisco, obedecendo aos raios de suas dimensões físicas. Dessa forma, foi possível definir quais discos classificados eram regidos pelos anéis interno e externo (Figura 2.2-D.1). Resolvendo a matriz da equação de rotação (Equação 2.44):

$$\begin{bmatrix} x_r(i) \\ y_r(i) \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.44)$$

onde $x_r(i)$ e $y_r(i)$ correspondem as coordenadas de cada pílula i estimada e x e y as coordenadas do disco de referência e $\theta_i = (2\pi)/i$ sendo $i = 1..n$ com n sendo o número de rótulos no disco.

Com o conhecimento da posição de apenas um disco no anel interno e um nó externo, foi possível estimar as posições dos demais discos, assim como, o centro dos 15 discos da armação denominada polisensidisco. No anel interno o índice i variou entre 1 e 5, enquanto que no anel externo esta variação foi de 1 a 10. A Figura 2.2-D.2 ilustra a

estimativa de todas as posições mencionadas.

A segunda etapa do AIA consiste em identificar as posições reais dos 15 discos, na armação dos antibiograma. Para tanto, são aplicadas 15 segmentações locais com dimensões de 100×100 pixels, na imagem da Figura 2.2-B, usando como centro as coordenadas das posições estimadas pela Equação 2.44.

Então, foi realizada uma correlação dentro de cada um das 15 sub-imagens segmentadas com um disco binário, com as dimensões físicas de um antibiograma: a posição correta da pílula foi identificada pelo par ordenado (x, y) onde ocorreu o valor de máxima correlação. A Figura 2.2-E ilustra todos os resultados das posições reais contidas na placa de Petri.

A definição da posição exata de cada pílula permitiu a execução da terceira etapa do AIA, na qual é identificado os rótulos de antibióticos e é determinada a região de inibição do antibióticos para cada um dos 15 discos.

A identificação dos rótulos dos antibióticos utiliza-se da comparação de rótulos de referência em relação aos rótulos segmentados por meio do algoritmo SIFT.

A identificação das regiões limitou-se a diâmetros d entre 10 e 29 mm, correspondentes ao mínimo e máximo usados na classificação dos discos presentes nos polisensidiscos da DME. Ressalta-se uma exceção da série Gram-positivo no diagnóstico da gonorreia, no qual o disco PEN10 apresenta halos com diâmetros maiores que 47 mm.

A informação desejada possui uma representação em baixas frequências, e corresponde ao vale formado pelos efeitos dos antibióticos. Assim, a imagem da Figura 2.2-E foi processada com um filtro passa-baixas do tipo FIR de ordem 25 e frequência de corte em 5% da largura de banda da imagem (Costa *et al.*, 2012b). Tal filtragem suavizou a imagem, tornando a transição entre amplitudes dos pixels menos brusca. Então cada região de interesse, contendo os 15 discos, é localmente segmentada por meio de um quadrado com 30 mm de lado.

2.4.2 Tesseract OCR

A ferramenta Tesseract OCR é um motor de reconhecimento ótico de caracteres de licença *Open Source* que foi desenvolvido pelo HP entre 1984 e 1994. Ela apareceu do nada para o UNLV – *Fourth Annual Test of OCR Accuracy* em 1995 (Rice *et al.*, 1995) e apresentou ótimos resultados e, em seguida, desapareceu de volta sob o mesmo sigilo sob as quais ela tinha sido desenvolvida (Smith, 2007). No final de 2005, a HP o disponibilizou com a licença *Open Source* e hoje ela está disponível em <http://code.google.com/p/tesseract-ocr>.

Essa ferramenta trabalha com uma tecnologia de análise de *layout* de páginas desenvolvida de forma independente (Mithe *et al.*, 2013). Assim, ela aceita imagens de entrada em formato binário (Mithe *et al.*, 2013) e pode lidar com tanto da forma tradicional preto sobre branco assim como o branco sobre o preto (Mithe *et al.*, 2013).

Em seu funcionamento interno, o processamento das imagens seguem uma linha de passo-a-passo tradicional, mas alguns dos estágios eram incomuns em seus dias e, possivelmente, permanece assim até agora (Smith, 2007).

O primeiro passo é uma análise de componentes conectados são armazenados (Smith, 2007; Mithe *et al.*, 2013). Nesta fase, os contornos são reunidos, puramente aninhando, em *blobs* (Smith, 2007; Mithe *et al.*, 2013). Estes *blobs* são organizados em linhas de texto e, estas linhas e regiões, são analisadas em passos fixos ou proporcionais ao texto (Smith, 2007; Mithe *et al.*, 2013). As linhas de texto são divididas em palavras de formas diferentes de acordo com o tipo de espaçamento entre os caracteres (Smith, 2007; Mithe *et al.*, 2013).

O reconhecimento prossegue, então, como um processo de dois passos. Na primeira passada, é feita uma tentativa para reconhecer cada palavra por vez (Smith, 2007; Mithe *et al.*, 2013). Cada palavra que teve um reconhecimento satisfatório é enviada para um classificador adaptativo como dados de treinamento (Smith, 2007; Mithe *et al.*, 2013). Desta maneira, este classificador adaptativo tem a chance de reconhecer o texto subsequente de forma mais precisa (Smith, 2007; Mithe *et al.*, 2013).

Desde que o classificador adaptativo pode ter aprendido algo útil tarde demais

para fazer uma contribuição perto início da página, uma segunda passada é executada nas palavras que não foram reconhecidas bem o suficiente na esperança que sejam reconhecidas desta vez (Smith, 2007).

A fase final resolve espaços escuros (relacionado ao contraste da imagem) e verifica hipóteses alternativas para para localizar textos pequenos (Smith, 2007).

3 METODOLOGIA

O presente trabalho foi realizado em quatro etapas distintas: Pesquisa Bibliográfica, Seleção de Materiais, Seleção de Ferramentas e a Proposta de Algoritmo de Identificação de Rótulos. Cada uma destas etapas será descrita em mais detalhes a seguir, sendo a última etapa a mais relevante e importante dentre elas.

Ao final das etapas citadas anteriormente, foram realizadas algumas simulações e experimentos a fim de investigar a efetividade do algoritmo proposto.

3.1 PESQUISA BIBLIOGRÁFICA

A pesquisa da base bibliográfica utilizada neste trabalho considerou a busca por livros, teses, monografias e artigos nas seguintes fontes especializadas: PubMed, *Institute of Electrical and Electronics Engineers* (IEEE), Springer, Association for Computing Machinery (ACM), Elsevier, Google Scholar, CiteSeer^X e trabalhos produzidos pelo projeto AUTOBAC. Esta busca visava o reconhecimento de caracteres em antibiogramas ou alguma técnica de auxílio a sua leitura.

O PubMed é uma base de dados que permite a pesquisa bibliográfica de artigos publicados em revistas de grande circulação da área médica. Ele foi desenvolvido pelo *National Center for Biotechnology Information* (NCBI), sendo mantido pela *National Library of Medicine* (NLM). Nesta fonte, foi realizada uma pesquisa com as palavras-chave “*antibiogram*”, “*identification*” e “*test*”, que retornou 882 trabalhos relacionados a este tema. Contudo, somente uma entre elas estava relacionada à automatização da identificação bacteriana, mas não se apresentou relevante ao tema proposto.

Outra combinação envolveu a troca da palavra-chave “*test*” por “*software*”. Este pesquisa resultou em 50 resultados diferentes e apresentou sete trabalhos associados ao tema. Entretanto, eles apresentam comparações entre ferramentas proprietárias (*software*) como o Vivek 1, Vivek 2 e Phoenix System. Em um deles, ele informa da eficiência do reconhecimento com espécies da *Enterobacteriaceae*, mas se apresenta

ineficiente no uso clínico geral (Carrol *et al.*, 2006).

Outra base bibliográfica pesquisada foi o IEEE. Embora a pesquisa tenha retornado cinco trabalhos com a palavra-chave “*antibiogram*”, os trabalhos de Legrand *et al.* (1995), Salgado *et al.* (2001) e Senyer e Efendiyev (2008) abordam apenas a identificação do halo, não os caracteres dos rótulos.

Nas bases de dados da Springer, Elsevier e da ACM, foi utilizada apenas a palavra-chave “*antibiogram*”, mas não houveram resultados.

O Google Scholar é uma ferramenta de pesquisa da empresa Google que permite pesquisar em trabalhos acadêmicos, literatura escolar, jornais de universidades e artigos variados. Nesta base de dados, a primeira pesquisa foi utilizada as palavras-chave “antibiograma” e “identificação”, que retornaram 2.820 resultados com 3 deles relevantes, mas já citados em outras fontes descritas acima. Já a pesquisa com as palavras-chave “*antibiogram*”, “*identification*” e “*automatic*” retornaram 3.760 resultados, mas nenhum relevante ao tema.

O CiteSeer^X é um motor de busca público, uma biblioteca digital e um repositório para artigos científicos com o foco primário em computação e em tecnologia da informação. Ele já recebeu patrocínio de empresas como Microsoft e NASA. Atualmente ele é patrocinado pela *The National Science Foundation* e possui mais de 4 milhões de documentos em seu acervo. Nesta fonte, foram utilizadas as palavras-chave “*antibiogram*” e “*identification*”, que retornou 709 resultados, mas nenhum relevante ao tema.

No Brasil, foram pesquisadas as bases de dados das instituições UnB, USP e IBICT. Todas as pesquisas neste âmbito utilizaram apenas a palavra-chave “antibiograma”. Na UnB, foram encontrados 11 resultados, mas somente a monografia de graduação de Costa (2014) foi considerada relevante, pois tratava da identificação automática de rótulos com o uso da técnica *Scale-Invariant Feature Transform* (SIFT). No entanto, a USP, com 330 resultados, e o IBICT, com 73 resultados, não trouxeram trabalhos relevantes.

3.2 SELEÇÃO DOS MATERIAIS

As imagens e o Algoritmo de Identificação de Antibiograma (AIA) utilizado neste trabalho foram cedidas pelo projeto AUTOBAC, desenvolvido no Laboratório de Instrumentação e Processamento de Imagens e Sinais (LIPIS), o qual faz parte dos Laboratórios de Engenharia & Inovação (LEI) da Faculdade UnB Gama.

A amostra cedida possui 32 imagens de placas de Petri com 6 discos (com o total de 192 rótulos) adquiridas e digitalizadas por Costa (2014) no Laboratório de Engenharia de Processos Ambiente e Energia da Faculdade de Engenharia da Universidade do Porto, Portugal, em 2013. Essas imagens foram compostas por placas de Petri de 90×15 mm contendo Ágar de Müller-Hinton (Oxoid Ltd., Hampshire, Reino Unido) e foi utilizado o instrumento ChemiDocTM MP System, da Bio-Rad no modo “*epi white illumination*” em conjunto com o *software* Image LabTM integrado ao sistema configurado para aquisições de imagens com tamanho de 1392×1040 pixels, 254 dpi e 8 bits. A Figura 3.1 apresenta um exemplo deste banco de imagens.

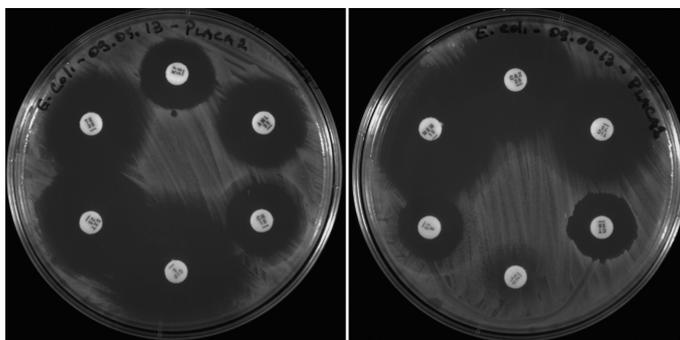


Figura 3.1: Exemplo de placas utilizadas no banco do padrão Oxoid.

Os rótulos das imagens seguem o padrão de disco descritos pelo padrão Oxoid: ML25, C Z30, CIP5, CN10, CT50, KF30, MEM10, RL25, S10, SXT25, TET30 e TIC75. O Apêndice A apresenta imagens com um exemplo de cada rótulo, as quais foram utilizadas nos grupos de controle para teste e comparação dos algoritmos.

3.3 SELEÇÃO DE FERRAMENTAS

Para a codificação e teste do algoritmo proposto e para a realização das comparações entre as diferentes técnicas foi utilizado um MacBook Pro com processador 2,2 GHz Intel Core i7 de 4 núcleos, memória RAM de 16 GiB 1600 MHz DDR3, placa de vídeo Intel

Iris Pro 1536 MiB e armazenamento SSD de 250 GiB. O sistema operacional instalado neste equipamento era o Mac OS X Yosemite, versão 10.10.2.

Os códigos do algoritmo foram escritos no *software* Matlab R2014b (8.4.0.150421, 64 bit). A escolha deste software se deu dada a maior familiaridade do orientador com ela, além do fato que o algoritmo AIA, que foi utilizado como comparativo neste trabalho, também fora escrito nesta ferramenta.

3.4 PROPOSTA DE ALGORITMO DE IDENTIFICAÇÃO DE RÓTULOS

A proposta de algoritmo de identificação dos rótulos também foi dividida em etapas distintas, a saber: segmentação dos rótulos, preparação das imagens, criação dos bancos de momentos e aplicação dos momentos invariantes ao AIA. Cada uma destas etapas está descrita nas próximas subseções.

3.4.1 Segmentação dos rótulos

Para que os rótulos pudessem ser identificados, eles devem ser pré-processados por meio de um algoritmo de segmentação. Essa segmentação seguiu o mesmo procedimento adotado por Costa (2014) (Figura 2.2-A a Figura 2.2-D). Este processo tem como objetivo separar os rótulos da imagem do antibiograma.

Após a segmentação, as imagens precisam, ainda, passar por novos pré-processamentos para que elas sejam utilizadas nas rotinas responsáveis pelo reconhecimento dos rótulos.

3.4.2 Preparação das imagens

Após o processo de segmentação do rótulo pelo AIA, as imagens sofreram um pré-processamento, com o intuito de remover os ruídos e informações que não estão relacionadas ao rótulo a ser identificado. Para isso, é calculado o histograma da imagem e, após isso, ela é binarizada.

A Figura 3.2 apresenta o fluxo para o pré-processamento da imagem.

O histograma das imagens foram baseados na distribuição de seus tons de cinza.

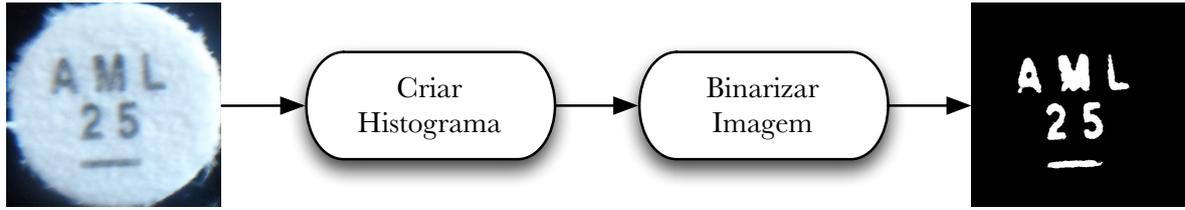


Figura 3.2: Fluxo de pré-processamento da imagem.

Assim, o histograma H para cada nível de cinza $i \in [0, 255]$ é dado pela Equação 3.1.

$$H(i) = \sum_x \sum_y \sigma_i(p(x, y)) \quad (3.1)$$

onde $p(x, y)$ é o nível de cinza do pixel localizado nas coordenadas (x, y) da imagem e σ_i é uma função definida na Equação 3.2.

$$\sigma_i(z) = \begin{cases} 1, & \text{se } z = i. \\ 0, & \text{caso contrário.} \end{cases} \quad (3.2)$$

Após esta etapa, a imagem é binarizada conforme o limiar de Otsu (Hu, 1962), sem levar em consideração as bordas escuras da imagem (Figura 3.2), pois elas não são significativas à determinação do rótulo. Desta forma, os quatro cantos da imagem foram substituídos pela cor branca (valor 255), para que elas não influenciem no cálculo de valor do momento do rótulo. Por fim, os valores foram invertidos para que o rótulo fique em branco e o fundo em preto.

Uma vez preparadas as imagens, tornou-se necessária a criação de um banco de momentos a partir do conjunto controle de imagens, que é utilizada posteriormente na identificação dos rótulos.

3.4.3 Criação do banco de momentos

Para o cálculo dos Momentos Invariantes, foi utilizado o script sugerido por Gonzalez *et al.* (2004) (Script B.1).

A Figura 3.3 apresenta o fluxo utilizado para a criação do banco de momentos.

O Script 3.1 apresenta os passos utilizados para a geração dos bancos de momentos

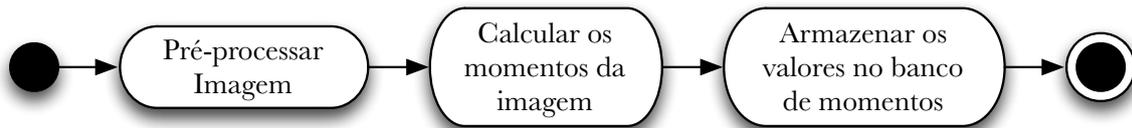


Figura 3.3: Fluxo de geração do banco de momentos invariantes.

e os scripts com a implementação das funções utilizadas neste processo encontram-se no Apêndice B.

Script 3.1: Script para a geração do banco de momentos.

```

1 listing = dir(fullfile('media/*'));
2
3 [length, a] = size(listing);
4 for files=1:length
5     name = listing(files).name;
6     if (name(1) ~= '.')
7         calculamedia(name);
8     end
9 end
  
```

3.4.4 Aplicação dos Momentos Invariantes ao AIA

O AIA possui vários passos para identificação do antibiograma. Entretanto, os Momentos Invariantes se aplicam apenas na identificação dos rótulos.

Desta forma, para que os Momentos Invariantes sejam aplicados ao AIA, a imagem do antibiograma precisa ser segmentada em rótulos individuais e, logo em seguida, binarizados (conforme apresentada na Subseção 3.4.2). Como a segmentação da imagem já é feita atualmente pelo AIA, o processo de aplicação deve ocorrer logo após essa segmentação. A Figura 3.4 apresenta um diagrama da integração dos Momentos Invariantes no AIA que se inicia após esta segmentação.

Todavia, antes que o processo de identificação dos rótulos ocorra, é necessária a criação de um banco de momentos inicial. Por isso, é necessário segmentar um ou mais rótulos de cada um dos tipos de forma aleatória e calcular os Momentos Invariantes e o desvio padrão desta amostra para que estes valores sejam utilizados como referência na identificação.

Após a criação do banco de momentos e das etapas de segmentação e binarização, é necessário computar cada um dos 7 Momentos Invariantes de cada um dos rótulos e,

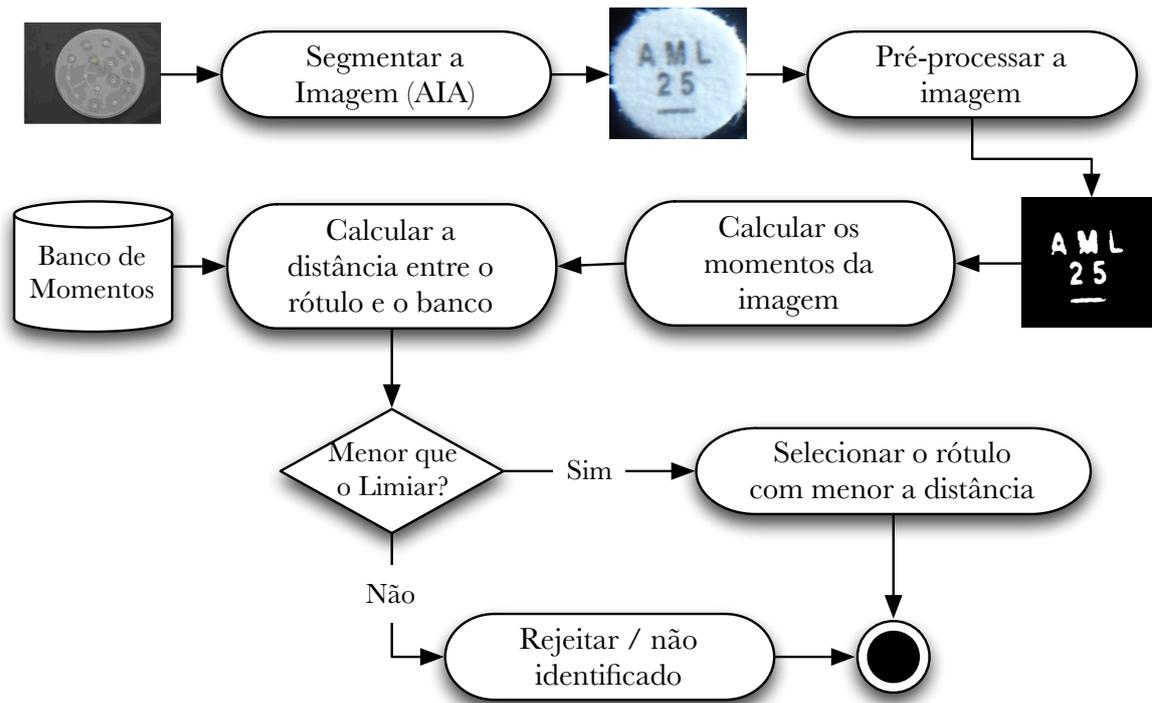


Figura 3.4: Diagrama de aplicação dos Momentos Invariantes ao AIA.

com base nestes valores de momentos, é calculado o valor da distância entre cada um dos momentos (características) com o valor do banco de momentos (características de referência).

Para o reconhecimento do rótulo foram utilizadas as propostas de Hu (1962) e Cash e Hatamian (1987). O primeiro utilizou a Distância Euclidiana entre os Momentos Centrais, enquanto o segundo utilizou várias combinações entre os diversos tipos de momentos e os comparou de várias formas como, por exemplo, a Distância Euclidiana e a Distância de Mahalanobis.

Ao final das comparações, é utilizado um limiar de aceitação mínimo (que depende de cada tipo de distância escolhido). Se dois ou mais rótulos estiverem dentro deste limiar, o menor entre eles é escolhido. Se nenhum valor estiver neste limiar, o rótulo é considerado como não identificado.

3.5 EXECUÇÃO DA SIMULAÇÃO E DOS EXPERIMENTOS

Inicialmente, os 192 rótulos foram segmentados pelo AIA e separados em 3 execuções de experimento, cada um com um número distinto de rótulos para o conjunto de controle

e experimental conforme apresentado na Tabela 3.1.

Tabela 3.1: Quantidade de rótulos separados por grupos considerando 12 rótulos distintos por grupo de controle. O experimento 1 possui 3 rótulos de cada tipo, o experimento 2 possui 4 rótulos de cada tipo e o experimento 3 possui 5 rótulos de cada tipo.

Experimento	Grupo de Controle	Grupo Experimental
1	36	156
2	48	144
3	60	132

Devida a necessidade de criação do banco de momentos e da quantidade de rótulos disponíveis para o experimento, foi utilizada a sugestão de Cash e Hatamian (1987) que sugeriu a divisão dos grupos de controle com 3, 4 e 5 itens quando se há uma amostra pequena para experimento.

Após a seleção dos 3 conjuntos de execução do experimento, toda a execução do experimento é executada com cada um destes grupos. A Figura 3.5 apresenta um diagrama de execução do experimento em cada um dos 3 grupos.

Inicialmente, foi gerado o banco de momentos do grupo de controle em separado do grupo experimental. Este passo é executado apenas uma vez e é utilizado o mesmo banco de momentos para todas as comparações durante a identificação.

Em seguida, as imagens do experimento são pré-processadas com o processo de segmentação e binarização. A partir deste ponto, existe uma atividade diferente para cada tipo de técnica utilizada para a comparação: Momento Central (Subseção 2.3.1), Momentos Invariantes (Subseção 2.3.2 e Subseção 2.3.3), SIFT (Subseção 2.2.2) e a ferramenta Tesseract OCR (Subseção 2.4.2), mas todos eles se iniciam da mesma forma.

Para permitir a execução do experimento, as imagens do grupo experimental foram armazenadas em um diretório chamado de *imagens*. Dentro deste diretório, as imagens foram armazenadas com o nome de seu rótulo para permitir a criação de um script que validasse o resultado da identificação de forma automática em relação ao método utilizado.

Ao final do experimento, foi executado um script que lê cada imagem dentro do diretório de *imagens* e executou a identificação conforme cada uma das técnicas es-

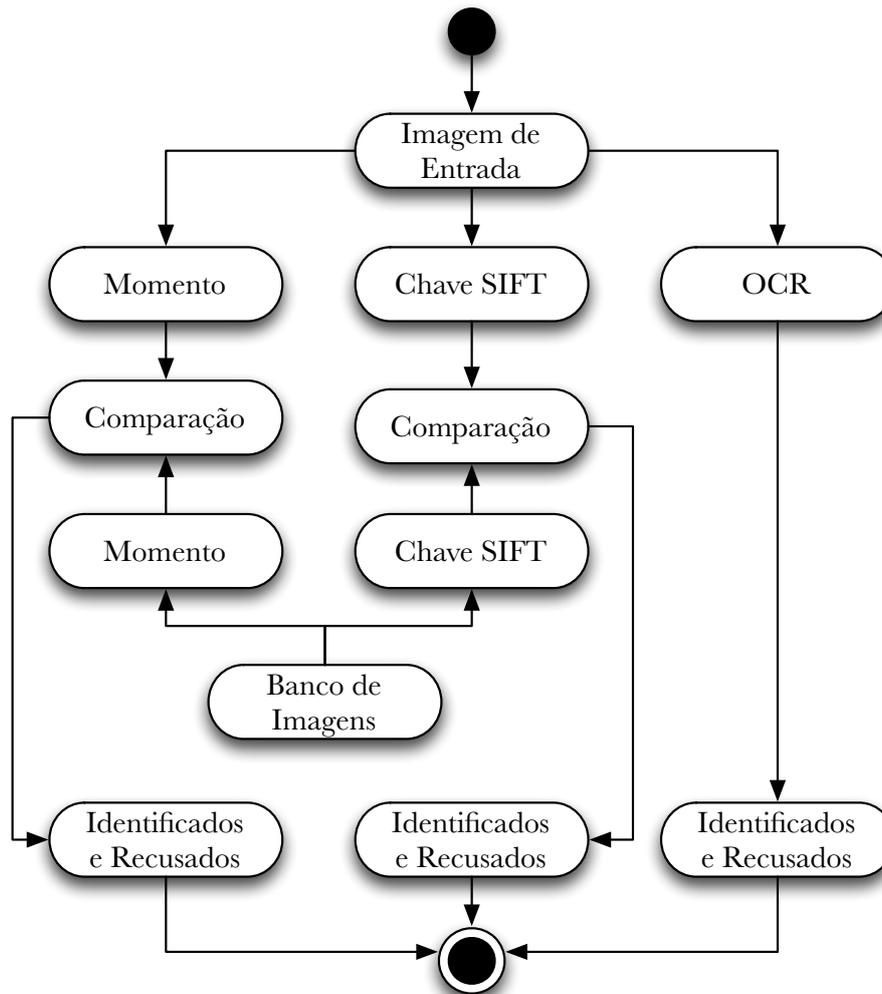


Figura 3.5: Diagrama de execução do experimento.

colhidas. Após este passo, o resultado da identificação foi validado conforme o nome original do arquivo da imagem e pôde ser identificado como “Identificado” (o rótulo foi identificado corretamente), “Recusado” (o rótulo informado é diferente do utilizado na comparação, ou seja, ele foi rejeitado corretamente), “Falso Positivo” (o resultado foi dado como verdadeiro, mas na verdade era falso) e “Falso Negativo” (o resultado foi dado como falso, mas na verdade era verdadeiro).

3.5.1 Geração do banco de momentos

Esse processo não se aplica a ferramenta Tesseract OCR.

Foi criado um diretório chamado de `media` e mais um diretório dentro dele contendo o nome do rótulo que ele representa. Este último diretório contém 3, 4 ou 5 arquivos de imagens dos rótulos com a extensão `.tiff`.

Após a criação dos diretórios que armazenam as imagens utilizadas na geração do banco de momentos, foi criado um script que lê tais diretórios em sequência e realiza os cálculos dos 7 Momentos Invariantes, o Momento Central e os desvios padrão de ambos. Estes resultados são armazenados nos diretórios `momentos` e `desvios`, dentro do diretório `banco`. Cada arquivo contém o nome do rótulo e a extensão de arquivos de valores `.mat` do Matlab.

3.5.2 Aplicação das medidas de comparação

No caso do Momento Central, as imagens foram comparadas com o banco de momentos (com o X , Y e seus desvios padrão σ_X e σ_Y) utilizando a Distância Euclidiana. Já com os Momentos Invariantes, eles foram comparados com o banco de momentos utilizando a Distância Euclidiana e a Distância de Mahalanobis (ambos com o uso do peso baseado no desvio padrão).

Segundo Cash e Hatamian (1987), os valores dos momentos se tornam mais sensíveis a ruídos a cada adição de nível na comparação. Assim foram utilizadas apenas os momentos invariantes ϕ_1 , ϕ_2 e ϕ_3 conforme sugestão deste autor.

De outro modo, como a técnica SIFT compara a similaridade entre imagens, foi utilizada as imagens que geraram o banco de momentos, conforme é proposto no AIA.

Para manter a similaridade com as outras técnicas, a ferramenta Tesseract OCR foi executada com a mesma imagem já pré-processada, ou seja, segmentada e binarizada.

3.5.3 Escolha do limiar de aceitação

Como os Momentos Invariantes, os Momentos Centrais e o SIFT utilizam-se de um valor de limiar de aceitação arbitrário e de difícil definição (pois ele é baseado na natureza da solução e na qualidade amostra utilizada), cada técnica utilizada neste trabalho possui um valor de limiar L diferente definido inicialmente como:

$$L_0 = \frac{F_{min} + F_{max}}{2} \quad (3.3)$$

onde L_0 é o valor do limiar inicial, F_{min} é o menor valor das características da amostra e F_{max} é o maior valor das características da amostra.

Após a definição do valor inicial, o experimento é executado em sequência e o valor de identificações corretas ou incorretas são avaliadas. Para cada uma dessas iterações, um novo valor de limar L_k foi escolhido para cada k -ésima iteração conforme a Equação 3.4.

$$L_k = L_{k-1} \times 0,9 \quad (3.4)$$

As iterações seguem com o intuito de reduzir os falsos positivos até que o seu valor estabilize ou até que o valor de identificações com sucesso comecem a diminuir. Neste caso, a iteração com o menor número de falsos positivos e o maior número de identificações com sucesso é escolhida.

4 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentados os resultados do experimento e, em seguida, uma discussão de seus significados. Também será apresentada uma correlação entre os resultados atuais (proposta de solução com os Momentos Invariantes) e os resultados da técnica utilizada pelo AIA (SIFT).

No início do experimento, o primeiro passo foi a geração do banco de momentos. Neste caso, as Tabelas 4.1 e 4.2 apresentam o banco de Momentos Invariantes com o uso de 3 rótulos aleatórios de cada tipo do padrão DME. A Tabela 4.3 apresenta o banco de Momentos Centrais dos mesmos 3 rótulos utilizado no banco de Momentos Invariantes. Vale ressaltar que o experimento também foi realizado com 4 e 5 rótulos de cada padrão para a formação do banco de momentos.

Tabela 4.1: Banco de Momentos Invariantes com o uso de 3 rótulos em sua formação (7 momentos invariantes).

Rótulo	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
ML25	0,255	$4,048 \times 10^3$	$5,836 \times 10^3$	$3,844 \times 10^4$	$7,141 \times 10^7$	$2,013 \times 10^5$	$-5,140 \times 10^7$
C Z30	0,320	$1,885 \times 10^3$	$1,621 \times 10^2$	$3,086 \times 10^5$	$2,318 \times 10^9$	$-1,179 \times 10^6$	$2,319 \times 10^8$
CIP5	0,546	$4,804 \times 10^2$	$1,413 \times 10^1$	$1,493 \times 10^2$	$1,076 \times 10^3$	$3,511 \times 10^3$	$-2,764 \times 10^4$
CN10	0,321	$2,371 \times 10^2$	$5,563 \times 10^3$	$1,446 \times 10^3$	$4,987 \times 10^6$	$2,201 \times 10^4$	$-2,010 \times 10^6$
CT50	0,403	$4,572 \times 10^2$	$1,398 \times 10^3$	$3,002 \times 10^4$	$-1,816 \times 10^7$	$5,747 \times 10^5$	$1,209 \times 10^7$
KF30	0,312	$2,728 \times 10^2$	$1,168 \times 10^3$	$1,405 \times 10^4$	$1,110 \times 10^7$	$1,620 \times 10^5$	$-3,995 \times 10^8$
MEM10	0,290	$8,335 \times 10^4$	$1,602 \times 10^2$	$5,985 \times 10^4$	$1,397 \times 10^6$	$-8,881 \times 10^6$	$-1,200 \times 10^6$
RL25	0,404	$5,504 \times 10^2$	$5,175 \times 10^3$	$1,297 \times 10^3$	$5,811 \times 10^6$	$2,497 \times 10^4$	$2,824 \times 10^7$
S10	0,431	$8,817 \times 10^2$	$6,684 \times 10^3$	$3,893 \times 10^4$	$-4,319 \times 10^7$	$-6,582 \times 10^5$	$-3,985 \times 10^8$
SXT25	0,362	$1,237 \times 10^2$	$1,895 \times 10^2$	$4,787 \times 10^4$	$1,083 \times 10^6$	$3,918 \times 10^5$	$1,415 \times 10^6$
TE30	0,333	$3,285 \times 10^2$	$1,189 \times 10^3$	$2,737 \times 10^4$	$6,057 \times 10^8$	$5,412 \times 10^6$	$-8,016 \times 10^8$
TIC75	0,340	$2,102 \times 10^2$	$9,989 \times 10^3$	$5,766 \times 10^4$	$4,410 \times 10^7$	$4,349 \times 10^5$	$-1,552 \times 10^6$

Naturalmente, interpretar uma comparação ou validação da solução computacional por meio das Tabelas 4.1, 4.2, 4.3, 4.4 e 4.5 é ineficaz. Nesse sentido, são apresentadas as Tabelas 4.6, 4.7 e 4.8, que objetivamente são constatadas as respostas efetivas de funcionalidade das técnicas em estudo.

Após o cálculo do banco de momentos, foi iniciada a execução do experimento com os Momentos Invariantes. Nesta comparação, é realizada uma comparação de distância entre um rótulo e os valores do banco de momentos. Desta forma, a Tabela 4.4 apresenta

Tabela 4.2: Banco de Momentos Invariantes com o uso de 3 rótulos em sua formação (desvio padrão).

Rótulo	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7
ML25	0,022	$1,237 \times 10^3$	$2,608 \times 10^3$	$4,413 \times 10^4$	$1,196 \times 10^6$	$3,070 \times 10^5$	$7,001 \times 10^7$
C Z30	0,029	$4,318 \times 10^4$	$4,771 \times 10^3$	$2,658 \times 10^5$	$1,127 \times 10^8$	$1,352 \times 10^6$	$3,568 \times 10^8$
CIP5	0,150	$1,874 \times 10^2$	$1,072 \times 10^1$	$1,634 \times 10^2$	$1,589 \times 10^3$	$4,019 \times 10^3$	$4,005 \times 10^4$
CN10	0,020	$1,050 \times 10^2$	$2,575 \times 10^3$	$1,166 \times 10^3$	$6,789 \times 10^6$	$2,347 \times 10^4$	$2,010 \times 10^6$
CT50	0,078	$1,599 \times 10^2$	$7,455 \times 10^4$	$2,547 \times 10^4$	$2,176 \times 10^7$	$6,138 \times 10^5$	$1,309 \times 10^7$
KF30	0,021	$6,373 \times 10^3$	$8,373 \times 10^4$	$2,217 \times 10^4$	$1,938 \times 10^7$	$2,945 \times 10^5$	$6,975 \times 10^8$
MEM10	0,025	$3,046 \times 10^4$	$4,403 \times 10^3$	$1,880 \times 10^4$	$1,178 \times 10^6$	$9,770 \times 10^6$	$5,972 \times 10^7$
RL25	0,081	$2,551 \times 10^2$	$4,491 \times 10^3$	$1,461 \times 10^3$	$9,289 \times 10^6$	$3,673 \times 10^4$	$8,315 \times 10^7$
S10	0,065	$3,095 \times 10^2$	$2,108 \times 10^3$	$1,090 \times 10^4$	$4,557 \times 10^7$	$1,007 \times 10^4$	$5,439 \times 10^7$
SXT25	0,058	$3,055 \times 10^3$	$7,812 \times 10^3$	$4,222 \times 10^4$	$1,135 \times 10^6$	$3,579 \times 10^5$	$1,726 \times 10^6$
TE30	0,016	$1,139 \times 10^3$	$4,293 \times 10^4$	$1,728 \times 10^4$	$2,167 \times 10^7$	$5,460 \times 10^5$	$1,481 \times 10^7$
TIC75	0,040	$4,699 \times 10^3$	$4,748 \times 10^3$	$3,072 \times 10^4$	$4,017 \times 10^7$	$2,696 \times 10^5$	$1,369 \times 10^6$

Tabela 4.3: Banco de Momento Central com o uso de 3 rótulos em sua formação.

Rótulo	X	Y	σ_X	σ_Y
ML25	38,32047	36,07609	2,63277	3,97189
C Z30	37,19463	34,83729	3,34203	2,07649
CIP5	36,22214	36,55227	1,38797	0,67810
CN10	39,98653	34,81200	4,59610	1,67827
CT50	34,72732	35,93553	1,83412	3,46737
KF30	36,49544	28,03636	7,21848	1,48453
MEM10	34,81204	38,09496	2,10217	3,05414
RL25	35,15217	37,24677	7,07154	3,21580
S10	33,94879	33,86213	7,66201	0,82337
SXT25	37,85711	32,23670	5,90072	3,81578
TE30	34,98987	36,34253	2,10899	2,56394
TIC75	39,12365	37,24601	0,57989	7,31144

Tabela 4.4: Momentos Invariantes de uma amostra aleatória do grupo experimental com 3 rótulos em sua formação.

Rótulo	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
ML25	0,308	$6,394 \times 10^3$	$1,139 \times 10^2$	$1,082 \times 10^3$	$3,270 \times 10^6$	$7,482 \times 10^5$	$-1,932 \times 10^6$
C Z30	0,290	$2,445 \times 10^3$	$1,113 \times 10^2$	$5,278 \times 10^7$	$2,758 \times 10^{11}$	$-2,006 \times 10^8$	$2,959 \times 10^{11}$
CIP5	0,471	$3,023 \times 10^2$	$7,254 \times 10^2$	$7,821 \times 10^3$	$1,863 \times 10^4$	$1,347 \times 10^3$	$2,828 \times 10^6$
CN10	0,227	$1,208 \times 10^2$	$1,045 \times 10^3$	$1,573 \times 10^4$	$5,323 \times 10^8$	$1,494 \times 10^5$	$-3,509 \times 10^8$
CT50	0,425	$5,101 \times 10^2$	$1,074 \times 10^3$	$3,548 \times 10^4$	$-1,132 \times 10^7$	$6,236 \times 10^5$	$1,875 \times 10^7$
KF30	0,290	$2,164 \times 10^2$	$9,662 \times 10^4$	$4,839 \times 10^6$	$9,039 \times 10^{11}$	$4,515 \times 10^7$	$3,183 \times 10^{10}$
MEM10	0,262	$6,443 \times 10^4$	$1,042 \times 10^2$	$1,735 \times 10^4$	$-8,022 \times 10^8$	$-3,099 \times 10^6$	$-2,191 \times 10^7$
RL25	0,381	$4,682 \times 10^2$	$1,948 \times 10^3$	$1,929 \times 10^4$	$8,556 \times 10^8$	$7,330 \times 10^6$	$-8,164 \times 10^8$
S10	0,455	$9,751 \times 10^2$	$1,134 \times 10^2$	$1,122 \times 10^3$	$-6,720 \times 10^7$	$3,679 \times 10^5$	$3,945 \times 10^6$
SXT25	0,268	$7,387 \times 10^3$	$5,818 \times 10^3$	$3,022 \times 10^4$	$6,404 \times 10^8$	$6,523 \times 10^6$	$3,956 \times 10^7$
TE30	0,262	$1,978 \times 10^2$	$1,610 \times 10^4$	$6,784 \times 10^5$	$6,560 \times 10^9$	$2,014 \times 10^6$	$-2,687 \times 10^9$
TIC75	0,337	$2,165 \times 10^2$	$8,255 \times 10^3$	$3,753 \times 10^4$	$2,549 \times 10^8$	$2,330 \times 10^5$	$-6,602 \times 10^7$

o exemplo de Momentos Invariantes de um único aleatório de cada tipo e a Tabela 4.5 apresenta os Momentos Centrais da mesma amostra.

Tabela 4.5: Momentos Centrais de uma amostra aleatória.

Rótulo	X	Y
ML25	34,77365	33,82770
C Z30	35,88834	38,96196
CIP5	37,22983	34,60147
CN10	33,35269	38,16431
CT50	38,31089	41,97708
KF30	42,00978	35,58513
MEM10	35,85837	32,58369
RL25	28,02446	35,85015
S10	40,61029	30,69853
SXT25	40,41628	45,24885
TE30	36,31973	36,61395
TIC75	33,05263	37,96370

Após a criação do banco de momentos, a geração dos Momentos Invariantes e a geração dos Momentos Centrais do grupo experimental, foi realizada a comparação do experimento.

No primeiro experimento, foram utilizados os Momentos Invariantes com a Distância Euclidiana (com o limiar $L = 0,87$) e com a Distância de Mahalanobis (com o limiar de $L = 50,5$). A Tabela 4.6 apresenta o resultado dessas comparações, considerando os itens identificados com sucesso (ID), os itens rejeitados com sucesso (R), os falsos positivos (FP) e os falsos negativos (FN). É notória a melhora no desempenho da técnica de Momentos Invariantes da Distância de Mahalanobis em relação a Distância Euclidiana para os limiares adotados, pois o uso da Distância de Mahalanobis apresentou um percentual de rótulos identificados com sucesso de 90,9% contra os 71,6% da Distância Euclidiana com o mesmo banco de momentos.

Por outro lado, a a Tabela 4.7 apresenta uma comparação entre o uso dos Momentos Invariantes com a Distância de Mahalanobis (com o limiar $L = 50,5$) em relação ao uso dos Momentos Centrais (com o limiar $L = 10,5$), conforme a sugestão de Hu (1962). Embora os Momentos Centrais sejam efetivos no reconhecimento de caracteres da imagem de um documento de texto, ele não se apresentou eficiente nos rótulos da amostra (com apenas 57% de sucesso na identificação), pois os rótulos possuem um alto índice de ruído e baixa resolução.

Tabela 4.6: Comparação entre os Momentos Invariantes com o uso da Distância Euclidiana ($L = 0,87$) em relação a Distância de Mahalanobis ($L = 50,5$) separados por itens identificados com sucesso (ID), rejeitados com sucesso (R), falsos positivos (FP) e falsos negativos (FN).

Rótulo	Distância Euclidiana						Distância de Mahalanobis					
	Acertos			Erros			Acertos			Erros		
	ID	R	Total	FP	FN	Total	ID	R	Total	FP	FN	Total
ML25	7	85	92	36	4	40	5	116	121	5	6	11
C Z30	12	129	141	36	3	39	10	161	171	4	5	9
CIP5	6	108	114	13	5	18	7	121	128	0	4	4
CN10	9	68	77	53	2	55	6	109	115	12	5	17
CT50	10	134	144	31	5	36	4	155	159	10	11	21
KF30	10	74	84	47	1	48	6	102	108	19	5	24
MEM10	14	145	159	20	1	21	9	162	171	3	6	9
RL25	15	108	123	57	0	57	11	142	153	23	4	27
S10	15	123	138	42	0	42	13	158	171	7	2	9
SXT25	10	78	88	43	1	44	7	118	125	3	4	7
TE30	11	93	104	28	0	28	7	110	117	11	4	15
TIC75	15	61	76	104	0	104	13	150	123	15	2	17
Total	134	1.206	1.340	510	22	532	98	1.604	1.702	112	58	170

Tabela 4.7: Comparação entre os Momentos Invariantes com o uso da Distância de Mahalanobis ($L = 50,5$) em relação ao Momento Central ($L = 10,5$) separados por itens identificados com sucesso (ID), rejeitados com sucesso (R), falsos positivos (FP) e falsos negativos (FN).

Rótulo	Momentos Invariantes						Momento Central					
	Acertos			Erros			Acertos			Erros		
	ID	R	Total	FP	FN	Total	ID	R	Total	FP	FN	Total
ML25	5	116	121	5	6	11	7	43	50	78	4	82
C Z30	10	161	171	4	5	9	10	96	106	69	5	74
CIP5	7	121	128	0	4	4	11	18	29	103	0	103
CN10	6	109	115	12	5	17	3	45	48	76	8	84
CT50	4	155	159	10	11	21	10	74	84	91	5	96
KF30	6	102	108	19	5	24	2	101	103	20	9	29
MEM10	9	162	171	3	6	9	8	93	101	72	7	79
RL25	11	142	153	23	4	27	1	137	138	28	14	42
S10	13	158	171	7	2	9	4	139	143	26	11	37
SXT25	7	118	125	3	4	7	1	118	119	3	10	13
TE30	7	110	117	11	4	15	11	25	36	96	0	96
TIC75	13	150	123	15	2	17	13	97	110	68	2	70
Total	98	1.604	1.702	112	58	170	81	986	1.067	730	75	805

Por último, a Tabela 4.8 apresenta a comparação dos Momentos Invariantes com o uso da Distância de Mahalanobis (com o limiar $L = 50,5$) com o limiar $L = 0,8$ em relação o uso da técnica SIFT (com o limiar $L = 10,5$). Neste caso, pode-se observar que a técnica SIFT apresentou um resultado ainda melhor que os Momentos Invariantes em conjunto com a Distância de Mahalanobis, pois ele obteve um resultado de identificação total 94,9%, ou seja, de 4% a mais do que a sua contrapartida.

No intuito de apresentar uma visualização melhor do desempenho de identificação

Tabela 4.8: Comparação entre os Momentos Invariantes com o uso da Distância de Mahalanobis ($L = 25$) com o SIFT ($L = 10,5$) separados por itens identificados com sucesso (ID), rejeitados com sucesso (R), falsos positivos (FP) e falsos negativos (FN).

Rótulo	Momentos Invariantes						SIFT					
	Acertos			Erros			Acertos			Erros		
	ID	R	Total	FP	FN	Total	ID	R	Total	FP	FN	Total
ML25	5	116	121	5	6	11	7	115	122	6	4	10
C Z30	10	161	171	4	5	9	14	159	173	6	1	7
CIP5	7	121	128	0	4	4	9	121	130	0	2	2
CN10	6	109	115	12	5	17	7	118	125	3	4	7
CT50	4	155	159	10	11	21	12	160	172	5	3	8
KF30	6	102	108	19	5	24	9	118	127	3	2	5
MEM10	9	162	171	3	6	9	12	163	175	2	3	5
RL25	11	142	153	23	4	27	15	144	159	21	0	21
S10	13	158	171	7	2	9	3	162	165	3	12	15
SXT25	7	118	125	3	4	7	11	117	128	4	0	4
TE30	7	110	117	11	4	15	11	112	123	9	0	9
TIC75	13	150	123	15	2	17	12	165	177	0	3	3
Total	98	1.604	1.702	112	58	170	122	1.654	1.776	62	34	96

das técnicas de Momento Central (M. Central), Momentos Invariantes com o uso da Distância Euclidiana (MI D_E), Momentos Invariantes com o uso da Distância de Mahalanobis (MI D_M) e o SIFT; a Figura 4.1 apresenta um gráfico com o percentual de acertos (rótulos identificados e rejeitados corretamente) e percentual de erros (falsos positivos e falsos negativos).

Embora nenhuma das técnicas de identificação apresentadas possuam um índice de identificação de 100%, eles ainda se apresentam aplicáveis ao método do AIA, pois este método já possui uma verificação de redundância de caracteres baseada na ordem e posicionamento dos discos (localização e ordem dos rótulos) (Costa, 2014, p. 53). Para isso, (Costa, 2014, p. 37-39) propôs uma técnica que verifica a quantidade de rótulos identificados em uma determinada sequência (de acordo com o padrão do disco utilizado) em todas as combinações possíveis. Ao final dessa comparação, o AIA escolhe aquela combinação que possui o maior índice de correspondência, o que torna a solução tolerante a pequenas falhas de identificação (falsos positivos ou falsos negativos).

Para que essa redundância funcione, é necessário que os antibiogramas de disco-difusão sigam um padrão de dispensação e rotulação (Costa, 2014). Se esta recomendação for seguida, em teoria, o AIA pode reconhecer corretamente os rótulos com qualquer técnica de reconhecimento de rótulos que possua uma taxa imediatamente superior a 50% de acertos.

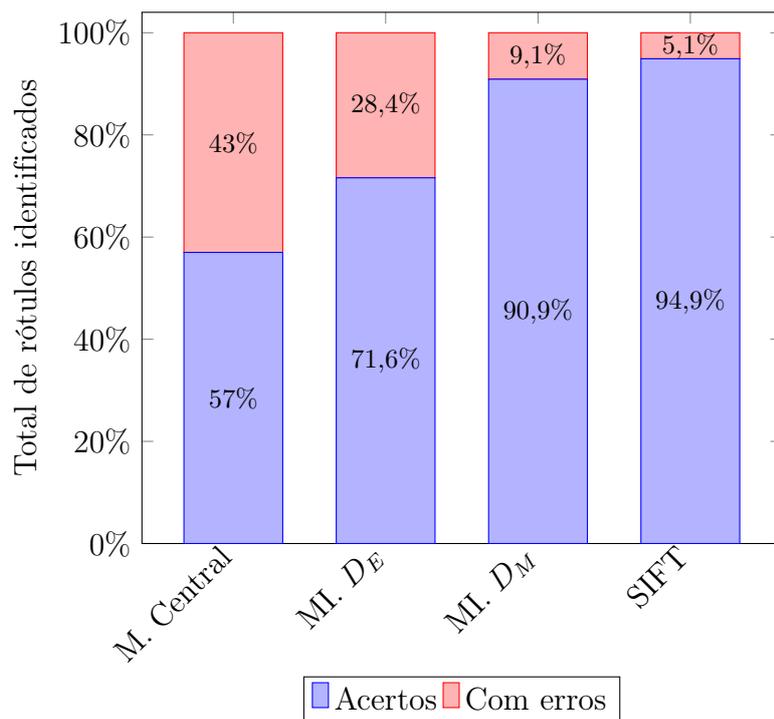


Figura 4.1: Total de rótulos identificados pelas técnicas: Momento Central (M. Central), Momento Invariante com o uso da Distância Euclidiana (MI. D_E), Momento Invariante com o uso da Distância de Mahalanobis (MI. D_M) e a técnica SIFT.

Em relação a ferramenta de Tesseract OCR, embora tenha sido realizados vários testes, ela foi incapaz de identificar quaisquer rótulos neste experimento com as imagens adquiridas. Alguns casos, ela foi capaz de identificar até dois caracteres corretamente, mas nenhum dos números de intensidade foi reconhecido. Acredita-se que a resolução disponível dos rótulos e a taxa de ruído seja grande o suficiente para que as técnicas utilizadas por essa ferramenta não tenha um bom desempenho no reconhecimento dos caracteres.

Após os experimentos relacionados a eficiência do reconhecimento dos rótulos, foi efetuada uma análise do tempo de execução das técnicas abordadas (apresentada na Tabela 4.9). Esta tabela apresenta o tempo de execução total de identificação da amostra de 156 rótulos do grupo experimental com o banco de momentos com 3 rótulos do grupo de controle (Tempo total), assim como o tempo médio de reconhecimento de um único rótulo (Tempo individual).

Para os Momentos Centrais e os Momentos Invariantes, o banco de momentos foi calculado uma única vez e seu tempo de execução não consta na Tabela 4.9. Na técnica SIFT, foi utilizado o mesmo algoritmo proposto pelo AIA, ou seja, são calculadas as

Tabela 4.9: Tempo de execução das técnicas aplicadas com o uso dos 156 rótulos do grupo experimental e 3 rótulos para o banco de momentos (em segundos).

Técnica	Tempo total	Tempo individual
Momento Central	10,6029	0,0679
Momentos Invariantes (Distância Euclidiana)	7,7844	0,0499
Momentos Invariantes (Distância de Mahalanobis)	8,0029	0,0513
SIFT	163,6450	1,0490

chaves SIFT das duas imagens e, em seguida, são comparadas para validar se existe uma correspondência (realizada com cada uma das 3 imagens do banco e escolhido o maior valor de semelhança entre elas para esta validação). Todavia, a Tabela 4.10 apresenta o tempo de execução (em segundos) das mesmas técnicas apresentadas, mas com a utilização de um banco de chaves SIFT, ou seja, com o uso da mesma abordagem da criação do banco de momentos utilizados nos Momentos Invariantes e Momentos Centrais.

Tabela 4.10: Tempo de execução das técnicas aplicadas com o uso dos 156 rótulos do grupo experimental com 3 rótulos para o banco de momentos e a utilização de banco de chaves SIFT (em segundos).

Técnica	Tempo total	Tempo individual
Momento Central	10,6029	0,0679
Momentos Invariantes (Distância Euclidiana)	7,7844	0,0499
Momentos Invariantes (Distância de Mahalanobis)	8,0029	0,0513
SIFT	40,9112	0,2622

Embora o tempo de execução seja menor com o uso da Distância Euclidiana, a diferença de desempenho na identificação dos rótulos não justifica a sua escolha. Pode-se perceber também que baseada na redundância de vizinhança de rótulos, no menor do tempo de execução dos Momentos Invariantes e na possibilidade de uso livre desta técnica (sem a necessidade de licenciamentos e *royalties*), o método proposto é considerado aplicável ao projeto AUTOBAC em conjunto com o AIA.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi proposta uma técnica de descritores baseada em Momentos Invariantes a fim de identificar rótulos alfanuméricos presentes em discos de antibiogramas por disco-difusão. Esta proposta visa auxiliar, também, o projeto AUTOBAC na sua necessidade de alteração da técnica comercial de reconhecimento de caracteres utilizada durante o seu desenvolvimento inicial.

Nos vários experimentos executados durante o desenvolvimento da solução proposta, o melhor caso apresentado foi a combinação do uso dos Momentos Invariantes com a Distância de Mahalanobis, que apresentou um percentual de identificação dos rótulos com sucesso superior a 90,9%. O tempo de execução da técnica proposta é considerado pequeno (aproximadamente 50 ms) mesmo quando utilizada uma ferramenta que necessita de muitos recursos do equipamento, o que torna atraente o estudo de sua aplicação em sistemas embarcados.

Dada a simplicidade da técnica e de implementação da proposta de uso dos Momentos Invariantes em conjunto com a Distância de Mahalanobis, existe a possibilidade de se portar o AIA, desenvolvido junto com o AUTOBAC, para dispositivos embarcados e, principalmente, permitir a criação de um equipamento e do uso comercial sem a necessidade de aquisição de licenças ou o pagamento de *royalties*.

Contudo, este processo de identificação não está preparado para a identificação de caracteres em imagens que contenham distorções afins (mal alinhadas, por exemplo). No contexto do projeto que este estudo está inserido, tal problema já seria evitado na etapa de aquisição, pois o projeto visa a construção de um equipamento de aquisição de imagens fixo e com configuração constante, o que evita distorções e diferenças significativas em relação ao posicionamento, alinhamento, iluminação e resolução das imagens.

5.1 TRABALHOS FUTUROS

Visando o andamento do projeto AUTOBAC, será efetuado um repasse da aplicação dos Momentos Invariantes no reconhecimento dos rótulos e a aplicação do método proposto no AIA. Outro ponto importante é a conversão do código-fonte de Matlab para a linguagem de programação C junto a biblioteca OpenCV, que foi definida para a construção dos equipamentos deste projeto. Por fim, é sugerida, também, a aplicação de técnicas de otimização no processo de reconhecimento de caracteres para viabilizar o seu uso em dispositivos embarcados.

REFERÊNCIAS BIBLIOGRÁFICAS

- Amato Neto, V., Nicodemo, A. C., e Lopes, H. V. (2007). *Antibióticos na prática médica*. Sarvier, São Paulo, 6 edition.
- Bauer, A. L., Kirby, W. M. M., Sherris, J. C., e Turck, M. (1966). *Antibiotic susceptibility testing by a standardized single disk method*. *Am. J. Clin. Pathol*, 45:493–496.
- Carrol, K. C., Glanz, B. D., Borek, A. P., Burger, C., Bhally, H. S., Henciak, S., e Flayhart, D. (2006). *Evaluation of the BD Phoenix Automated Microbiology System for Identification and Antimicrobial Susceptibility Testing of Enterobacteriaceae*. *Journal of Clinical Microbiology*, 44:3506–3509.
- Cash, G. L. e Hatamian, M. (1987). *Optical character recognition by the method of moments*. *Computer Vision, Graphics, and Image Processing*, 39(3):291–310.
- Costa, L. A., de Andrade, M. M., Costa Jr., E. A., e Costa, L. F. R. (2014). *Identificação automática de caracteres em antibiogramas com o uso de momentos invariantes*. In *XXIV Congresso Brasileiro de Engenharia Biomédica – CBEB*, Anais do Congresso Brasileiro de Engenharia Biomédica, Uberlândia.
- Costa, L. F. R. (2014). *Método automático para identificação da região de inibição e de rótulos alfanuméricos de antibióticos posicionados em antibiogramas por disco-difusão*. Monografia (bacharelado em engenharia eletrônica), Universidade de Brasília, Brasília.
- Costa, L. F. R., de Andrade, M. M., Costa Jr., E. A., e Espinoza, B. L. M. (2012a). *Método automático para identificação de rótulos alfanuméricos de antibióticos posicionados em antibiograma*. In *XIX Congresso Brasileiro de Automática*, Anais do Congresso Brasileiro de Automática, Campina Grande.
- Costa, L. F. R., de Andrade, M. M., Jr., E. A. C., e Espinoza, B. L. M. (2012b). *Método automático para identificação robusta da região de inibição em antibiogramas*. In *Congresso Brasileiro de Engenharia Biomédica - CBEB*, Anais do Congresso Brasileiro de Engenharia Biomédica, Porto de Galinhas - PE.

- Courvalin, P. (1996). *Interpretive reading of in vitro antibiotic susceptibility tests (the antibiogramme)*. *ASM News*, 2:S26–185.
- De Maesschalck, R., Jouan-Rimbaud, D., e Massart, D. L. (2000). *The Mahalanobis Distance*. *Chemometrics and intelligent laboratory systems*, 50(1):1–18.
- Flusser, J. (2000). *On the independence of rotation moment invariants*. *Pattern Recognition*, 33(9):1405 – 1410.
- Flusser, J. e Suk, T. (2006). *Rotation Moment Invariants for Recognition of Symmetric Objects*. *IEEE Transactions on Image Processing*, 15.
- Gonzalez, R., Woods, R. E., e Eddins, S. L. (2004). *Digital Image Processing Using MATLAB*. Prentice Hall, 1 edition.
- Gonzalez, R. C. e Woods, R. E. (2002). *Digital image processing*. Prentice Hall.
- Hadjigeorgiou, K., Kastanos, E., Kyriakides, A., e Pitris, C. (2009). *Raman spectroscopy for UTI diagnosis and antibiogram*. In *Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference on*, pages 1–4.
- Hu, M. K. (1962). *Visual pattern recognition by moment invariants*. *Information Theory, IRE Transactions on*, 8(2):179–187.
- Jain, A. K., Duin, R. P. W., e Mao, J. (2000). *Statistical pattern recognition: a review*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37.
- Legrand, L., Aho, L., e Dusserre, L. (1995). *Antibiogram automatic reading with a micro-computer based image processing system*. In *Engineering in Medicine and Biology Society, 1995 and 14th Conference of the Biomedical Engineering Society of India. An International Meeting, Proceedings of the First Regional Conference., IEEE*, pages 2/30–2/31.
- Lowe, D. G. (1999). *Object recognition from local scale-invariant features*. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE.
- Lowe, D. G. (2004). *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*. US Patent 6,711,293.

- Mahalanobis, P. C. (1936). *On the generalized distance in statistics. Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55.
- Mithe, R., Indalkar, S., e Divekar, N. (2013). *Optical character recognition. International Journal of Recent Technology and Engineering (IJRTE) Volume*, 2:72–75.
- Mori, S., Nishida, H., e Yamada, H. (1999). *Optical character recognition*. John Wiley & Sons, Inc.
- Organick, E. I. (1966). *A Fortran IV primer*. Addison-Wesley.
- Otsu, N. (1979). *A Threshold Selection Method from Gray-Level Histograms. Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66.
- Rice, S. V., Jenkins, F. R., e Nartker, T. A. (1995). *The fourth annual test of OCR accuracy*. Technical report, Technical Report 95.
- Salgado, L., Menendez, J., Rendon, E., Garcia, N., e Ruiz, V. (2001). *Automatic antibiograms inhibition halo determination through texture and directional filtering analysis. In Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 629–632 vol.2.
- Senyer, N. e Efendiyev, C. (2008). *Automatic antibiogram inhibition zone diameter determination through circular hough transform. In Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th*, pages 1–4.
- Smith, R. (2007). *An overview of the Tesseraact OCR engine. In icdar*, pages 629–633. IEEE.
- Tavares, W. (2006). *Antibióticos e Quimioterápicos para o clínico*. Atheneu, São Paulo, 1 edition.

Apêndices

A RÓTULOS UTILIZADOS NO GRUPO DE CONTROLE

A Figura A.1 apresenta um exemplo de cada um dos tipos de rótulo do padrão DME utilizados no grupo de controle.

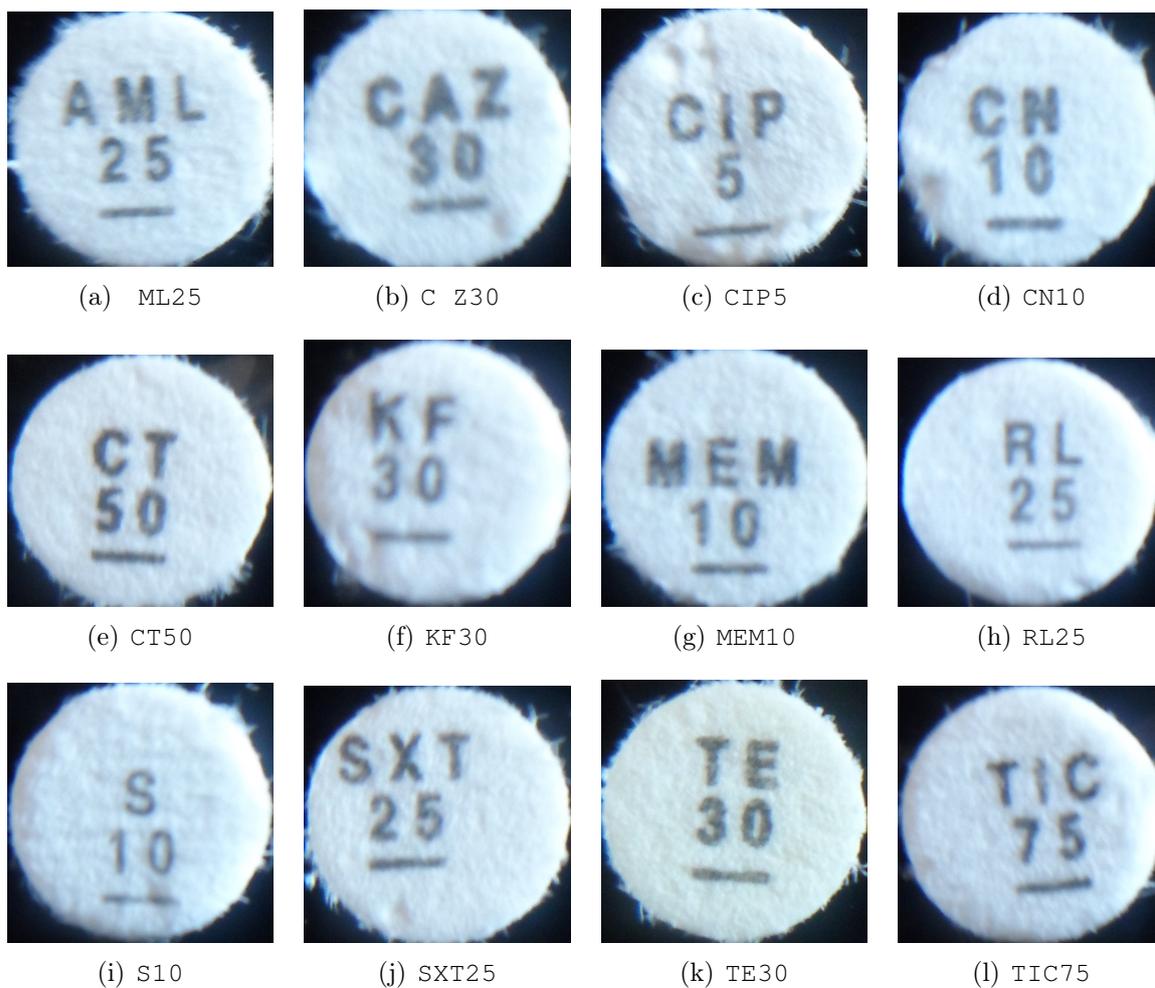


Figura A.1: Exemplo de rótulos do padrão DME utilizados para o grupo de controle.

B SCRIPTS DO ALGORITMO DE IDENTIFICAÇÃO DE RÓ- TULOS

O Script B.1 apresenta a função sugerida por Gonzalez *et al.* (2004) para o cálculo dos Momentos Invariantes.

Script B.1: Função responsável pelo cálculo dos Momentos Invariantes.

```
1 function phi = invmoments(F)
2 %INVMOMENTS Compute invariant moments of image.
3 % PHI = INVMOMENTS(F) computes the moment invariants of the image
4 % F. PHI is a seven-element row vector containing the moment
5 % invariants as defined in equations (11.3-17) through (11.3-23) of
6 % Gonzalez and Woods, Digital Image Processing, 2nd Ed.
7 %
8 % F must be a 2-D, real, nonsparse, numeric or logical matrix.
9
10 % Copyright 2002-2004 R. C. Gonzalez, R. E. Woods, & S. L. Eddins
11 % Digital Image Processing Using MATLAB, Prentice-Hall, 2004
12 % $Revision: 1.5 $ $Date: 2003/11/21 14:39:19 $
13
14 if (ndims(F) ~= 2) | issparse(F) | ~isreal(F) | ~(isnumeric(F) | ...
15                                     islogical(F))
16     error(['F must be a 2-D, real, nonsparse, numeric or logical ' ...
17           'matrix.']);
18 end
19
20 F = double(F);
21 phi = compute_phi(compute_eta(compute_m(F)));
22
23 %-----%
24 function m = compute_m(F)
25
26 [M, N] = size(F);
27 [x, y] = meshgrid(1:N, 1:M);
28
29 % Turn x, y, and F into column vectors to make the summations a bit
30 % easier to compute in the following.
31 x = x(:);
32 y = y(:);
33 F = F(:);
34
35 % DIP equation (11.3-12)
36 m.m00 = sum(F);
37 % Protect against divide-by-zero warnings.
38 if (m.m00 == 0)
39     m.m00 = eps;
40 end
41 % The other central moments:
42 m.m10 = sum(x .* F);
43 m.m01 = sum(y .* F);
```

```

44 m.m11 = sum(x .* y .* F);
45 m.m20 = sum(x.^2 .* F);
46 m.m02 = sum(y.^2 .* F);
47 m.m30 = sum(x.^3 .* F);
48 m.m03 = sum(y.^3 .* F);
49 m.m12 = sum(x .* y.^2 .* F);
50 m.m21 = sum(x.^2 .* y .* F);
51
52 %-----%
53 function e = compute_eta(m)
54
55 % DIP equations (11.3-14) through (11.3-16).
56
57 xbar = m.m10 / m.m00;
58 ybar = m.m01 / m.m00;
59
60 e.eta11 = (m.m11 - ybar*m.m10) / m.m00^2;
61 e.eta20 = (m.m20 - xbar*m.m10) / m.m00^2;
62 e.eta02 = (m.m02 - ybar*m.m01) / m.m00^2;
63 e.eta30 = (m.m30 - 3 * xbar * m.m20 + 2 * xbar^2 * m.m10) / m.m00^2.5;
64 e.eta03 = (m.m03 - 3 * ybar * m.m02 + 2 * ybar^2 * m.m01) / m.m00^2.5;
65 e.eta21 = (m.m21 - 2 * xbar * m.m11 - ybar * m.m20 + ...
66           2 * xbar^2 * m.m01) / m.m00^2.5;
67 e.eta12 = (m.m12 - 2 * ybar * m.m11 - xbar * m.m02 + ...
68           2 * ybar^2 * m.m10) / m.m00^2.5;
69
70 %-----%
71 function phi = compute_phi(e)
72
73 % DIP equations (11.3-17) through (11.3-23).
74
75 phi(1) = e.eta20 + e.eta02;
76 phi(2) = (e.eta20 - e.eta02)^2 + 4*e.eta11^2;
77 phi(3) = (e.eta30 - 3*e.eta12)^2 + (3*e.eta21 - e.eta03)^2;
78 phi(4) = (e.eta30 + e.eta12)^2 + (e.eta21 + e.eta03)^2;
79 phi(5) = (e.eta30 - 3*e.eta12) * (e.eta30 + e.eta12) * ...
80         ( (e.eta30 + e.eta12)^2 - 3*(e.eta21 + e.eta03)^2 ) + ...
81         (3*e.eta21 - e.eta03) * (e.eta21 + e.eta03) * ...
82         ( 3*(e.eta30 + e.eta12)^2 - (e.eta21 + e.eta03)^2 );
83 phi(6) = (e.eta20 - e.eta02) * ( (e.eta30 + e.eta12)^2 - ...
84         (e.eta21 + e.eta03)^2 ) + ...
85         4 * e.eta11 * (e.eta30 + e.eta12) * (e.eta21 + e.eta03);
86 phi(7) = (3*e.eta21 - e.eta03) * (e.eta30 + e.eta12) * ...
87         ( (e.eta30 + e.eta12)^2 - 3*(e.eta21 + e.eta03)^2 ) + ...
88         (3*e.eta12 - e.eta30) * (e.eta21 + e.eta03) * ...
89         ( 3*(e.eta30 + e.eta12)^2 - (e.eta21 + e.eta03)^2 );

```

A função apresentada no Script B.2 foi criada para auxiliar nas comparações do Momento Central.

Script B.2: Função responsável pelo cálculo do Momento Central.

```

1 function m = compute_m(F)
2
3 [M, N] = size(F);
4 [x, y] = meshgrid(1:N, 1:M);

```

```

5
6 % Turn x, y, and F into column vectors to make the summations a bit
7 % easier to compute in the following.
8 x = x(:);
9 y = y(:);
10 F = F(:);
11
12 % DIP equation (11.3-12)
13 m.m00 = sum(F);
14 % Protect against divide-by-zero warnings.
15 if (m.m00 == 0)
16     m.m00 = eps;
17 end
18 % The other central moments:
19 m.m10 = sum(x .* F);
20 m.m01 = sum(y .* F);
21 m.m11 = sum(x .* y .* F);
22 m.m20 = sum(x.^2 .* F);
23 m.m02 = sum(y.^2 .* F);
24 m.m30 = sum(x.^3 .* F);
25 m.m03 = sum(y.^3 .* F);
26 m.m12 = sum(x .* y.^2 .* F);
27 m.m21 = sum(x.^2 .* y .* F);
28
29 end

```

Em relação ao pré-processamento da imagem (histograma e binarização), foi criada a função conforme o Script B.3.

Script B.3: Função responsável pelo pré-processamento da imagem do rótulo.

```

1 function nova = prepara(img)
2
3 try
4     im = rgb2gray(img);
5 catch
6     im = img;
7 end
8
9 [x, y]=size(im);
10
11 B = zeros(size(im));
12 meio = mean2(im);
13
14 centrox = x/2;
15 centroy = y/2;
16 menor = min(centrox, centroy) * 0.8;
17
18 [xtotal, ytotal]=size(B);
19 for y=1:ytotal
20     for x=1:xtotal
21         d = sqrt((centrox - x)^2 + (centroy - y)^2);
22         if (d > menor)
23             im(x, y) = meio;
24             B(x,y)=255;
25         end

```

```

26     end
27 end
28 L = graythresh(im);
29
30 B = im2bw(im, L*.9);
31 for y=1:yttotal
32     for x=1:xttotal
33         d = sqrt((centrox - x)^2 + (centroy - y)^2);
34         if (d > menor)
35             B(x,y)=255;
36         end
37     end
38 end
39 B = ~B;
40 nova = B;
41 end

```

Em seguida, o Script B.4 apresenta a função responsável pelo cálculo e armazenamento da média dos Momentos Invariantes e Momento Central, assim como os seus valores de desvios padrão.

Script B.4: Função responsável pelo cálculo dos detalhes do banco de momentos.

```

1 function calculamedia(rotulo)
2 listing = dir(fullfile(strcat('media/', rotulo, '/*.tiff')));
3
4 phi1 = [];
5 phi2 = [];
6 phi3 = [];
7 phi4 = [];
8 phi5 = [];
9 phi6 = [];
10 phi7 = [];
11 x = [];
12 y = [];
13
14 [length, a] = size(listing);
15 for files=1:length
16     name = listing(files).name;
17     name = strcat('media/', rotulo, '/', name);
18
19     im_rgb=imread(name);
20     B = prepara(im_rgb);
21
22     phi = invmoments(B);
23
24     m = compute_m(B);
25
26     x = horzcat(x, m.m10 / m.m00);
27     y = horzcat(y, m.m01 / m.m00);
28
29     phi1 = horzcat(phi1, phi(1));
30     phi2 = horzcat(phi2, phi(2));
31     phi3 = horzcat(phi3, phi(3));
32     phi4 = horzcat(phi4, phi(4));

```

```

33     phi5 = horzcat(phi5, phi(5));
34     phi6 = horzcat(phi6, phi(6));
35     phi7 = horzcat(phi7, phi(7));
36 end
37
38 d1 = std(phi1);
39 d2 = std(phi2);
40 d3 = std(phi3);
41 d4 = std(phi4);
42 d5 = std(phi5);
43 d6 = std(phi6);
44 d7 = std(phi7);
45 dx = std(x);
46 dy = std(y);
47
48 phi1 = mean(phi1);
49 phi2 = mean(phi2);
50 phi3 = mean(phi3);
51 phi4 = mean(phi4);
52 phi5 = mean(phi5);
53 phi6 = mean(phi6);
54 phi7 = mean(phi7);
55 x = mean(x);
56 y = mean(y);
57
58 rotulo
59 save(strcat('banco/media/', rotulo), 'x', 'y', 'phi1', 'phi2', 'phi3', '
    phi4', 'phi5', 'phi6', 'phi7');
60 save(strcat('banco/desvio/', rotulo), 'dx', 'dy', 'd1', 'd2', 'd3', 'd4',
    'd5', 'd6', 'd7');
61
62 end

```

Para gerar as tabelas de resultados das comparações, foi criado um script para cada tipo de comparação. Assim, o Script B.5 apresenta o cálculo dos Momentos Centrais, o Script B.6 apresenta o cálculo dos Momentos Invariantes com o uso da Distância Euclidiana e, por último, o Script B.7 apresenta os Momentos Invariantes com o uso da Distância de Mahalanobis.

Script B.5: Função responsável pelo cálculo de comparações do Momento Central.

```

1 id = containers.Map();
2 rj = containers.Map();
3 fp = containers.Map();
4 fn = containers.Map();
5
6 listing = dir(fullfile('banco/media/*.mat'));
7 [length, a] = size(listing);
8
9 anterior = 10000;
10 for files=1:length
11     name = listing(files).name;
12     nameend = size(name);
13     banco = name(1:nameend(2)-4);

```

```

14
15 load(strcat('banco/media/', banco));
16 load(strcat('banco/desvio/', banco));
17
18 try
19     id(banco);
20 catch
21     id(banco) = 0;
22     rj(banco) = 0;
23     fp(banco) = 0;
24     fn(banco) = 0;
25 end
26
27 listing_img = dir(fullfile('imagem'));
28 [length2, a] = size(listing_img);
29 for geral=1:length2
30     dirs = listing_img(geral).name;
31     if (dirs(1) ~= '.')
32         images = dir(fullfile(strcat('imagem/', dirs, '/*.tiff')));
33         [length3, b] = size(images);
34         for index=1:length3
35             name = images(index).name;
36             nameend = size(name);
37             rot = name(1:nameend(2)-5);
38
39             im_rgb=imread(strcat('imagem/', dirs, '/', rot, '.tiff'));
40             B = prepara(im_rgb);
41             phin = invmoments(B);
42
43             mn = compute_m(B);
44             xn = mn.m10 / mn.m00;
45             yn = mn.m01 / mn.m00;
46             md = (dx + dy)/2;
47             d = sqrt(dx/md*(x - xn)^2+dy/md*(y - yn)^2);
48
49             try
50                 id(rot);
51             catch
52                 id(rot) = 0;
53                 rj(rot) = 0;
54                 fp(rot) = 0;
55                 fn(rot) = 0;
56             end
57
58             if (d > 4.5)
59                 if (strcmp(banco, rot))
60                     fn(rot) = fn(rot) + 1;
61                 else
62                     rj(rot) = rj(rot) + 1;
63                 end
64             else
65                 if (strcmp(banco, rot))
66                     id(rot) = id(rot) + 1;
67                 else
68                     fp(rot) = fp(rot) + 1;
69                 end
70             end
71         end
end

```

```

72         end
73     end
74 end
75
76 keys(id)
77 values(id)
78 values(rj)
79 values(fp)
80 values(fn)

```

Script B.6: Função responsável pelo cálculo de comparações dos Momentos Invariantes com o uso da Distância Euclidiana.

```

1 id = containers.Map();
2 rj = containers.Map();
3 fp = containers.Map();
4 fn = containers.Map();
5
6 listing = dir(fullfile('banco/media/*.mat'));
7 [length1, a] = size(listing);
8
9 anterior = 10000;
10 for files=1:length1
11     name = listing(files).name;
12     nameend = size(name);
13     banco = name(1:nameend(2)-4);
14
15     load(strcat('banco/media/', banco));
16     load(strcat('banco/desvio/', banco));
17
18     try
19         id(banco);
20     catch
21         id(banco) = 0;
22         rj(banco) = 0;
23         fp(banco) = 0;
24         fn(banco) = 0;
25     end
26
27 listing_img = dir(fullfile('imagem'));
28 [length2, a] = size(listing_img);
29 for geral=1:length2
30     dirs = listing_img(geral).name;
31     if (dirs(1) ~= '.')
32         images = dir(fullfile(strcat('imagem/', dirs, '/*.tiff')));
33         [length3, b] = size(images);
34         for index=1:length3
35             name = images(index).name;
36             nameend = size(name);
37             rot = name(1:nameend(2)-5);
38
39             im_rgb=imread(strcat('imagem/', dirs,'/', rot, '.tiff'));
40             B = prepara(im_rgb);
41             phin = invmoments(B);
42
43             media_desvio = (d1+d2+d3) / 3;
44             d = sqrt(d1/media_desvio*(sign(phin)*log(phin) - ...
45                 sign(phin(1))*log(phin(1)))^2 + ...

```

```

46         d2/media_desvio*(sign(phi2)*log(phi2) - ...
47         sign(phin(2))*log(phin(2)))^2 + ...
48         d3/media_desvio*(sign(phi3)*log(phi3) - ...
49         sign(phin(3))*log(phin(3)))^2);
50
51     try
52         id(rot);
53     catch
54         id(rot) = 0;
55         rj(rot) = 0;
56         fp(rot) = 0;
57         fn(rot) = 0;
58     end
59
60     if (d > 0.87)
61         if (strcmp(banco, rot))
62             fn(rot) = fn(rot) + 1;
63         else
64             rj(rot) = rj(rot) + 1;
65         end
66     else
67         if (strcmp(banco, rot))
68             id(rot) = id(rot) + 1;
69         else
70             fp(rot) = fp(rot) + 1;
71         end
72     end
73 end
74 end
75 end
76 end
77
78 keys(id)
79 values(id)
80 values(rj)
81 values(fp)
82 values(fn)

```

Script B.7: Função responsável pelo cálculo de comparações dos Momentos Invariantes com o uso da Distância de Mahalanobis.

```

1 id = containers.Map();
2 rj = containers.Map();
3 fp = containers.Map();
4 fn = containers.Map();
5
6 listing = dir(fullfile('banco/media/*.mat'));
7 [length1, a] = size(listing);
8
9 anterior = 10000;
10 for files=1:length1
11     name = listing(files).name;
12     nameend = size(name);
13     banco = name(1:nameend(2)-4);
14
15     load(strcat('banco/media/', banco));
16     load(strcat('banco/desvio/', banco));
17

```

```

18     try
19         id(banco);
20     catch
21         id(banco) = 0;
22         rj(banco) = 0;
23         fp(banco) = 0;
24         fn(banco) = 0;
25     end
26
27     listing_img = dir(fullfile('imagem'));
28     [length2, a] = size(listing_img);
29     for geral=1:length2
30         dirs = listing_img(geral).name;
31         if (dirs(1) ~= '.')
32             images = dir(fullfile(strcat('imagem/', dirs, '/*.tiff')));
33             [length3, b] = size(images);
34             for index=1:length3
35                 name = images(index).name;
36                 nameend =size(name);
37                 rot = name(1:nameend(2)-5);
38
39                 im_rgb=imread(strcat('imagem/',dirs, '/', rot, '.tiff'));
40                 B = prepara(im_rgb);
41                 phin = invmoments(B);
42
43                 media_desvio = (d1+d2+d3) / 3;
44                 d = sqrt(((sign(phin)*log(phin) - ...
45                     sign(phin(1))*log(phin(1)))^2)/media_desvio^2 + ...
46                     ((sign(phin(2))*log(phin(2)) - ...
47                     sign(phin(2))*log(phin(2)))^2)/media_desvio^2 + ...
48                     ((sign(phin(3))*log(phin(3)) - ...
49                     sign(phin(3))*log(phin(3)))^2)/media_desvio^2);
50
51                 try
52                     id(rot);
53                 catch
54                     id(rot) = 0;
55                     rj(rot) = 0;
56                     fp(rot) = 0;
57                     fn(rot) = 0;
58                 end
59                 if (d > 50.5)
60                     if (strcmp(banco, rot))
61                         fn(rot) = fn(rot) + 1;
62                     else
63                         rj(rot) = rj(rot) + 1;
64                     end
65                 else
66                     if (strcmp(banco, rot))
67                         id(rot) = id(rot) + 1;
68                     else
69                         fp(rot) = fp(rot) + 1;
70                     end
71                 end
72             end
73         end
74     end
75 end

```

```

76
77 keys(id)
78 values(id)
79 values(rj)
80 values(fp)
81 values(fn)

```

Para a execução das comparações com o uso da técnica SIFT, foi utilizado os scripts de exemplo do Lowe (1999), disponível em: <http://www.cs.ubc.ca/~lowe/keypoints/>. Dos scripts disponibilizados por Lowe (1999), apenas o arquivo `match.m` foi alterado para que não houvesse a pausa no reconhecimento e não apresentasse a imagem de comparação ao final do processo (necessário para que houvesse o reconhecimento automatizado da amostra experimental).

Assim, a Script B.8 apresenta o arquivo `match.m` alterado conforme indicado e o Script B.9 apresenta o cálculo da identificação com o uso da técnica SIFT.

Script B.8: Arquivo de comparação (*match*) do SIFT com a remoção da pausa e apresentação da imagem final.

```

1  % num = match(image1, image2)
2  %
3  % This function reads two images, finds their SIFT features, and
4  %   displays lines connecting the matched keypoints.   match is accepted
5  %   only if its distance is less than distRatio times the distance to the
6  %   second closest match.
7  % It returns the number of matches displayed.
8  %
9  % Example: match('scene.pgm', 'book.pgm');
10
11 function num = match(image1, image2)
12
13 % Find SIFT keypoints for each image
14 [im1, des1, loc1] = sift(image1);
15 [im2, des2, loc2] = sift(image2);
16
17 % For efficiency in Matlab, it is cheaper to compute dot products between
18 %   unit vectors rather than Euclidean distances. Note that the ratio of
19 %   angles (acos of dot products of unit vectors) is a close approximation
20 %   to the ratio of Euclidean distances for small angles.
21 %
22 % distRatio: Only keep matches in which the ratio of vector angles from
23 %   the nearest to second nearest neighbor is less than distRatio.
24 distRatio = 0.6;
25
26 % For each descriptor in the first image, select its match to second
27 %   image.
28 des2t = des2'; % Precompute matrix transpose
29 for i = 1 : size(des1,1)
30     dotprods = des1(i,:) * des2t; % Computes vector of dot products

```

```

30     [vals,indx] = sort(acos(dotprods)); % Take inverse cosine and sort
      results
31
32     % Check if nearest neighbor has angle less than distRatio times 2nd.
33     if (vals(1) < distRatio * vals(2))
34         match(i) = indx(1);
35     else
36         match(i) = 0;
37     end
38 end
39
40 num = sum(match > 0);

```

Script B.9: Função responsável pelo cálculo de comparações utilizando a técnica SIFT.

```

1  id = containers.Map();
2  rj = containers.Map();
3  fp = containers.Map();
4  fn = containers.Map();
5
6  listing = dir(fullfile('media/*'));
7  [length_a, a] = size(listing);
8
9  for files=1:length_a
10     banco = listing(files).name;
11
12     if (banco(1) == '.')
13         continue;
14     end
15
16     listing_img = dir(fullfile('imagem'));
17     [length, a] = size(listing_img);
18     for geral=1:length
19         dir = listing_img(geral).name;
20         if (dir(1) ~= '.')
21             images = dir(fullfile(strcat('imagem/', dir, '/*.tiff')));
22             [length, b] = size(images);
23             for index=1:length
24                 name = images(index).name;
25                 nameend = size(name);
26                 nameimg = name(1:nameend(2)-5);
27
28                 im_rt = strcat('imagem/', dir, '/', nameimg, '.tiff');
29                 im_b = dir(fullfile(strcat('media/', banco, '/*.tiff')));
30
31                 [length_b, a] = size(im_b);
32                 d = 0;
33                 for bn=1:length_b
34                     arq_banco = im_b(bn).name;
35                     if (arq_banco(1) == '.')
36                         continue;
37                     end
38
39                     nome_banco = strcat('media/', banco, '/', arq_banco);
40                     d_match = match(nome_banco, im_rt);
41                     if (d_match > d)
42                         d = d_match;
43                     end

```

```
44         end
45
46     try
47         id(nameimg);
48     catch
49         id(nameimg) = 0;
50         rj(nameimg) = 0;
51         fp(nameimg) = 0;
52         fn(nameimg) = 0;
53     end
54
55     if (d < 8)
56         if (strcmp(banco, nameimg))
57             fn(nameimg) = fn(nameimg)+1;
58         else
59             rj(nameimg) = rj(nameimg) + 1;
60         end
61     else
62         if (strcmp(banco, nameimg))
63             id(nameimg) = id(nameimg) + 1;
64         else
65             fp(nameimg) = fp(nameimg)+1;
66         end
67     end
68
69     end
70
71     end
72 end
73
74 keys(id)
75 values(id)
76 values(rj)
77 values(fp)
78 values(fn)
```
