



Universidade Federal de Pernambuco
CIn - Centro de Informática

Mestrado Profissional em Ciência da Computação

**Um método para classificação de imagens
de madeira usando *Local Binary Patterns***

Alex Paulo Alves de Oliveira

Dissertação de Mestrado

Recife
12 de Março de 2013

Universidade Federal de Pernambuco
CIn - Centro de Informática

Alex Paulo Alves de Oliveira

Um método para classificação de imagens de madeira usando
Local Binary Patterns

Trabalho apresentado ao Programa de Mestrado Profissional em Ciência da Computação do CIn - Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Prof. Dr. George Darmiton da Cunha Cavalcanti*

Recife
12 de Março de 2013

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Oliveira, Alex Paulo Alves de

Um método para classificação de imagens de madeira usando local binary patterns. / Alex Paulo Alves de Oliveira. - Recife: O Autor, 2013.

xiv, 127 folhas: fig., tab., gráf., quadro

Orientador: George Darmiton da Cunha Cavalcanti.

Dissertação (mestrado profissional) - Universidade Federal de Pernambuco. Cln, Ciência da Computação, 2013.

Inclui bibliografia e apêndice.

1. Inteligência artificial. 2. Processamento de imagem. 3. Reconhecimento de padrão. 4. Reconhecimento de madeira. I. Cavalcanti, George Darmiton da Cunha (orientador). II. Título.

006.3

CDD (23. ed.)

MEI2013 – 060

Dissertação de Mestrado Profissional apresentada por **Alex Paulo Alves de Oliveira** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título, "**Um método para classificação de imagens de madeira usando Local Binary Patterns**", orientada pelo Professor **George Darmiton da Cunha Cavalcanti** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

Prof. José Osman dos Santos
Instituto Federal de Sergipe

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 12 de março de 2013.

Prof^ª. EDNA NATIVIDADE DA SILVA BARROS
CoordenadorA da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedico este trabalho a meus pais e à minha esposa.

Agradecimentos

Agradeço à Deus e à minha família.

Resumo

O tráfico ilegal de madeiras é um problema no Brasil, percebido com mais frequência nas alfândegas da Amazônia. O objetivo desse trabalho é o desenvolvimento de um método para classificação de imagens de madeira. As imagens, usadas nessa pesquisa, foram fornecidas pela Embrapa e pelo VRI (UFPR).

Para o classificador criado, cada imagem é representada pelo histograma resultante da aplicação do operador *LBP* (*Local Binary Patterns*). A classificação desenvolvida tem como base o aprendizado baseado em instâncias, utilizando o algoritmo *K-NN* (*K-Nearest Neighbor*).

O aumento na quantidade de amostras, disponíveis para um mesmo teste, foi suficiente para tornar mais evidentes as diferenças de performance entre as diversos cenários elaborados.

Foram consideradas duas abordagens de *Cross-Validation*: O *K-Fold Cross-Validation* e o *Leave-One-Out Cross-Validation*. Quase sempre, quando o *Leave-One-Out Cross-Validation* foi adotado, os resultados apresentam uma acurácia melhor em relação à outra abordagem.

Neste trabalho, também foram realizados alguns testes para mensurar a robustez em relação ao ruído, e, ficou constatado que o ruído pode influenciar os resultados da classificação.

A normalização influenciou os resultados obtidos pelo classificador, entretanto, dentre as variáveis consideradas, essa foi a menos influente.

Foi possível perceber que a métrica adotada, para mensurar distâncias, influencia elementos importantes: o índice de acertos e a velocidade de resposta (processamento computacional exigido). O *Kullback Leibler Divergence* foi a métrica que apresentou melhores resultados.

O classificador construído neste trabalho se mostrou igualmente eficiente para bases com imagens homogêneas (com mesma dimensão e formato) e heterogêneas;

Palavras-chave: Aprendizado de Máquina, Aprendizado baseado em instância, Classificação, Imagem, Madeira

Abstract

The illegal timber traffic is a problem in Brazil, it is seen more often in customs Amazon. The aim of this work is to develop a method for classifying images of wood. The images used in this research was provided by Embrapa and the *VRI* (UFPR).

For the classifier created, each image is represented by the histogram resulting from the application of the operator *LBP* (*Local Binary Patterns*). The classification is based on instance based learning, using the algorithm *K-NN* (*K-Nearest Neighbor*).

The increase in the number of samples available for the same test, was sufficient to render most obvious differences in performance between the various scenarios prepared.

We considered two approaches of Cross-Validation: The *K-Fold Cross-Validation* and *Leave-One-Out Cross-Validation*. Almost always, when the *Leave-One-Out Cross-Validation* was adopted, the results show a better accuracy compared to other approach.

In this work, some tests were also performed to measure the allusive tolerance to noise, and it was found that noise can affect the results of the classification.

It is possible to say that normalization influenced the results obtained by the classifier, however, among the variables considered, this was the least influential.

It could be observed that the metric adopted to measure distances, influence two important elements: the hit rate and response speed (computational processing required). The *Kullback Leibler Divergence* is the metric that showed better results.

The classifier built in this work was equally efficient for bases that has images with the same size and format, and for other kinds of base that hasn't.

Keywords: Machine Learning, Instance Based Learning, Classification, Image, Wood

Lista de Figuras

1.1	Mapa exibindo as fronteiras da Amazônia (Fonte: Fragmento da imagem vista em http://www.brasil-turismo.com/mapas/mapa-politico.htm)	3
1.2	Mapa exibindo a Pan-Amazônia (Fonte: (SANTOS, 2010, p. 29))	5
1.3	Gráfico do desmatamento (Fonte: (SANTOS, 2010, p. 40))	6
2.1	Foto macro de uma mosca (Fonte: http://www.cursodefotografia-gratis.com)	11
2.2	Olho humano	18
2.3	Imagens de uma criança (Fonte: (SONKA; HLAVAC; BOYLE, 2007, p. 7-8))	19
2.4	Histograma	22
2.5	Histograma de intensidade de pixel (Fonte: Adaptado de (ODOM, 2006, p. 5))	23
2.6	Histogramas (Fonte: Adaptado de (TEO, 2003, p. 26))	24
2.7	Aplicação do LBP (Fonte: Adaptado de (PIETIKÄINEN et al., 2011, p. 14))	25
2.8	Passos <i>LBP</i> (Fonte: Adaptado de (PETROU; SEVILLA, 2006, p. 574))	26
2.9	<i>LBP</i> para macro textura (Fonte: Adaptado de (PETROU; SEVILLA, 2006, p. 575))	28
2.10	Simulando ambiente de inspeção de madeira (Fonte: (NISKANEN, 2003, p. 105))	30
2.11	LBP para reconhecimento de face (Fonte: (PIETIKÄINEN et al., 2011, p. 152))	31
2.12	LBP para detecção de olhos (Fonte: (PIETIKÄINEN et al., 2011, p. 154))	32
2.13	LBP para detecção de face (Fonte: (PIETIKÄINEN et al., 2011, p. 157))	32
2.14	LBP para discernir emoção (Fonte: (PIETIKÄINEN et al., 2011, p. 164))	33
2.15	Variações de K no K-NN (Fonte: Adaptado de (FERRERO, 2009, p. 33))	42
2.16	<i>Distância Euclidiana x Distância de Manhattan</i> (Fonte: (PUTTER, 2010, p. 36))	43
2.17	Diagrama de Voronoi	43
2.18	Variação do diagrama de <i>Voronoi</i> em função da métrica (Fonte: http://en.wikipedia.org/wiki/Voronoi_diagram)	44
2.19	Resultado da execução do <i>K-NN</i> (Fonte: Adaptado do <i>help</i> do <i>Matlab</i>)	45
2.20	<i>K-Fold Cross-Validation</i> com $K = 3$	49

2.21	Imagens de Cedro com ruído gerado pelo <i>Matlab</i> (Fonte: Adaptação de imagens fornecidas pela <i>Embrapa</i>)	51
2.22	Resultado de inserção de ruído no <i>Matlab</i> (Fonte: Adaptado do <i>help</i> do <i>Matlab</i> - http://www.mathworks.com/help/toolbox/images/ref/imnoise.html)	51
2.23	Sistema para reconhecimento de madeira (Fonte: (KHALID et al., 2008, p. 11))	53
3.1	Fluxo geral do método de classificação de madeira proposto	56
3.2	Micrótomo; Cortes em madeira (Fonte: <i>Embrapa</i> ; Adaptação de http://www.pauleiraguitars.com/materias/tipos-de-cortes-de-madeira)	58
3.3	Corte em acácia; Zoom no corte da acácia (Fonte: Adaptação de http://pt.wikipedia.org/wiki/Ficheiro:Acacia_koaia_wood.jpg)	60
3.4	Efeito da luminosidade (Fonte: Adaptado de http://artesatividades.blogspot.com.br/2011/04/luzsombrapropriaesombraprojetada_09.html)	61
3.5	Vizinhanças circulares (Fonte: Adaptado de (PIETIKÄINEN et al., 2011, p. 14))	62
3.6	Histograma calculado à partir de uma amostra	65
3.7	Efeitos da Normalização	66
3.8	Etapa 1 - Construção do banco de dados das imagens	67
3.9	Etapa 2 - Construção da base de conhecimento	68
3.10	Etapa 3 - Realizando predições	69
3.11	Passos da Etapa 3	70
4.1	Câmera utilizada (Fonte: (EMBRAPA, 1972))	72
4.2	Outras lupas (Fonte: (EMBRAPA, 1972))	73
4.3	Imagens de Cedro e Faveira (Fonte: (EMBRAPA, 1972))	77
4.4	Imagens de Goiabão e Louro Faia (Fonte: (EMBRAPA, 1972))	77
4.5	Comparação entre Andiroba e Goiabão (Fonte: Adaptação de imagens fornecidas pela <i>Embrapa</i>)	88

Lista de Quadros

2.1	Conjunto de treinamento com suas instâncias, atributos e classes	38
4.1	Amostras utilizadas para a primeira bateria de testes	76

Lista de Tabelas

2.1	Exemplo de distribuição de frequência numérica por faixa	20
2.2	Exemplo de distribuição de frequência numérica por valor	20
2.3	Exemplo de distribuição de frequência por categoria	21
4.1	Resumo quantitativo das imagens desta pesquisa	71
4.2	Resultado da primeira bateria de testes	76
4.3	Todas as imagens fornecidas pela <i>Embrapa</i>	78
4.4	Etapa 1 - Ruído no conjunto de validação (<i>K-Fold Cross-Validation</i>)	79
4.5	Etapa 1 - Ruído no conjunto de validação (<i>Leave-One-Out Cross-Validation</i>)	80
4.6	Etapa 2 - Ruído em todas instâncias (<i>K-Fold Cross-Validation</i>)	81
4.7	Etapa 2 - Ruído em todas instâncias (<i>Leave-One-Out Cross-Validation</i>)	81
4.8	Etapa 3 - Ruído em parte instâncias (<i>K-Fold Cross-Validation</i>)	83
4.9	Etapa 3 - Ruído em parte das instâncias (<i>Leave-One-Out Cross-Validation</i>)	85
4.10	Testando EKLD (<i>K-Fold Cross-Validation</i>)	89
4.11	Testando EKLD (<i>Leave-One-Out Cross-Validation</i>)	90
4.12	Melhores resultados da quinta bateria de testes (<i>K-Fold Cross-Validation</i>)	92
4.13	Melhores resultados da quinta bateria de testes (<i>Leave-One-Out Cross-Validation</i>)	93
4.14	Resumo dos resultados dos testes com as imagens da UFPR	97
A.1	Resultados da quinta bateria de testes (<i>K-Fold Cross-Validation</i>)	122
A.2	Resultados da quinta bateria de testes (<i>Leave-One-Out Cross-Validation</i>)	125

Lista de Gráficos

4.1	Teste inserindo ruído apenas no conjunto de validação	80
4.2	Teste inserindo ruído em todas as instâncias	82
4.3	Inserindo ruído em parte das instâncias (<i>K-Fold Cross-Validation</i>)	84
4.4	Inserindo ruído em parte das instâncias (<i>Leave-One-Out Cross-Validation</i>) . . .	86
4.5	Teste mesclando imagens da Embrapa e imagens da UFPR	94
4.6	Teste com todas as 2240 imagens da UFPR	98

Sumário

1	INTRODUÇÃO	1
1.1	Motivação	2
1.1.1	Contextualização	4
1.2	Objetivo	7
1.3	Código fonte	8
1.4	Estrutura deste trabalho	8
2	REVISÃO DE LITERATURA	10
2.1	Referencial teórico	10
2.1.1	Imagem digital	10
2.1.2	Textura	11
2.1.3	Visão computacional	13
2.1.4	Etapas comuns no processamento digital de imagens	17
2.1.5	Distribuição de frequência	20
	Histogramas	22
	Histograma de uma imagem	23
2.1.6	<i>LBP - Local Binary Patterns</i>	25
	Deixando o LBP invariante quanto à rotação	26
	Tornando o LBP apropriado para macro-texturas	27
	Como calcular o contraste de uma vizinhança LBP	28
	Aplicações de LBP no trato com madeira	29
	Aplicações frequentes do LBP	30
2.1.7	Normalização	34
	Normalização para a “norma de um”	35
	Normalização Z-scores	35
2.1.8	<i>Instance Based Learning</i>	37
	KNN (K-Nearest Neighbor)	39
2.1.9	<i>Cross-Validation</i>	46

Objetivos do Cross-Validation	46
Resubstitution Validation	47
Hold-Out Validation	48
K-Fold Cross-Validation	48
Leave-One-Out Cross-Validation	49
Repeated K-Fold Cross-Validation	50
2.1.10 Ruído em imagens digitais	50
2.2 Soluções alternativas	51
2.3 Trabalhos relacionados	53
3 MÉTODO PROPOSTO	56
3.1 Processos comuns entre as etapas	57
3.1.1 Criação da imagem digital	57
Seleção	57
Preparação	58
Digitalização	59
3.1.2 Extração de características	61
Aplicação do LBP	62
Cálculo dos histogramas	64
Normalização	65
3.2 Etapa 1 - Construção do banco de dados das imagens	67
3.3 Etapa 2 - Construção da base de conhecimento	68
3.4 Etapa 3 - Realizando predições	69
4 EXPERIMENTOS	71
4.1 Banco de dados das imagens	71
4.1.1 Ferramentas utilizadas	72
4.1.2 Criação da imagem digital	73
Seleção	74
Preparação	74
Digitalização	74
4.2 Resultados	75
4.2.1 Teste com parte das imagens da Embrapa e sem usar Cross-Validation	75
4.2.2 Teste com todas as imagens da Embrapa, usando Cross-Validation	77
4.2.3 Teste com todas as imagens da Embrapa, adicionando ruído	79

4.2.4	Teste com todas as imagens da Embrapa, usando novas métricas de distância	87
4.2.5	Teste mesclando imagens de bancos distintos (Embrapa e UFPR)	91
4.2.6	Teste com todas as imagens da UFPR	96
4.3	Análise	100
4.3.1	Dificuldades encontradas	101
4.3.1.1	<i>K-NN</i> do <i>Matlab</i> (<i>knnclassify</i>) não mensurava distâncias pela Equação 4.1	101
4.3.1.2	Qualidade das imagens utilizadas	102
4.3.2	Limitações e análise de validade	104
4.3.2.1	Das limitações	104
4.3.2.2	Da validade dos métodos utilizados	105
4.3.3	Sobre a normalização	105
4.3.4	Sobre as abordagens de <i>Cross-Validation</i>	106
4.3.5	Sobre as métricas de distância	107
5	CONCLUSÕES	109
6	TRABALHOS FUTUROS	112
6.1	Estudar outras métricas de distância	112
6.2	Realizar novos testes com outro descritor de texturas	112
6.3	Ajustar a qualidade das imagens em função das situações reais	113
6.4	Implementação de uma versão para <i>Smart Phone</i>	114
6.5	Criar vários classificadores	114
A	RESULTADOS DA QUINTA BATERIA DE TESTES	121

CAPÍTULO 1

INTRODUÇÃO

As técnicas de processamento digital de imagens são utilizadas para inúmeros objetivos, como: facilitar a interpretação de imagens radiográficas e outras imagens utilizadas na indústria, na medicina e nas ciências biológicas; geógrafos estudam padrões de poluição a partir de imagens aéreas de satélite; na arqueologia, métodos de processamento de imagens têm restaurado imagens fotográficas borradas satisfatoriamente (GONZALEZ; WOODS, 2010, p. 4).

Segundo Gonzalez e Woods (2010, p. 5), entre os problemas típicos de aprendizagem de máquina, que costumam utilizar técnicas de processamento de imagens, é possível citar:

- Reconhecimento automático de caracteres;
- Sistema de visão computacional industrial para inspeção e montagem de produtos, análise de impressões digitais, análise de imagens radiográficas e amostras de sangue;
- Processamento de imagens aéreas e de satélites para previsão do tempo e análise de degradação do ambiental.

A madeira é um material naturalmente heterogêneo que apresenta grande diversidade em seus aspectos macroscópicos (por exemplo, na cor e na textura) e microscópicos (por exemplo, nas dimensões das células), nas propriedades físicas (densidade, retração, e outras), mecânicas (compressão, flexão e outras) e na durabilidade, das quais resultam diferentes perspectivas tecnológicas, isto é, diferentes possibilidades de aplicação. Esta diversidade de madeiras parece aumentar em regiões tropicais (Bessa, 2009, p. 1).

Graças às características inerentes a cada tipo específico de madeira, e às técnicas de processamento digital de imagens, combinadas com técnicas de aprendizado de máquina, todas amplamente divulgadas na academia, é possível pensar num método automatizado para classificação das madeiras através de suas imagens digitais.

Essa análise automatizada será o resultado da combinação de um pré-processamento para extração de características (abordado na Seção 3.1.2), realizado nas imagens extraídas das madeiras, com técnicas clássicas de aprendizagem de máquina.

Assim, esta dissertação deve subsidiar a futura automatização do processo de classificação de madeiras tropicais, oriundas da Amazônia, tendo como insumos as imagens digitais das madeiras. Esses insumos foram disponibilizados por:

- *Embrapa Amazônia Oriental* (<http://www.cpatu.embrapa.br>), que é uma das unidades da *Embrapa* (*Empresa Brasileira de Pesquisa Agropecuária* - <http://www.embrapa.br>), que fica alocada dentro da própria floresta amazônica, e será referenciada, daqui por diante, apenas como *Embrapa*;
- *VRI* (Centro de pesquisa em *Visão Robótica Imagem* - <http://web.inf.ufpr.br/vri>), localizado na *UFPR* (*Universidade Federal do Paraná* - <http://www.ufpr.br>).

Adicionalmente, a *Embrapa* enviou imagens referentes às madeiras sobre as quais incidem os maiores índices de tráfico ilegal, nas alfândegas que fazem a circunscrição da floresta amazônica.

1.1 Motivação

A sistematização da informação sobre as madeiras das diferentes espécies, incluindo aspectos de caracterização, identificação e utilização, tem sido desde sempre considerada importante, e existem publicadas obras de referência para diversos tipos de madeiras, agrupadas por regiões ou grupos de famílias (Bessa, 2009, p. 1). A preocupação com a sustentabilidade das florestas, principalmente nas regiões tropicais, e a necessidade de respeitar convenções e acordos internacionais, têm majorado significativamente a necessidade de caracterização/identificação de amostras de madeira (Bessa, 2009, p. 2-5).

Adicionalmente, com a verificação da entrada, no mercado madeireiro, de madeiras provenientes de abates ilegais. A disponibilização dessa informação tem se mostrado importante para entidades públicas e privadas, notadamente para o *IBAMA*, para a *Polícia Federal* e para a *Embrapa*. Sem falar na indústria legalizada de fornecimento de madeiras, que sofre o ônus de uma concorrência ilegal e desprovida de sustentabilidade, conforme dados informais disponibilizados pela *Embrapa*.

Em geral, o comércio de madeiras tropicais restringe-se a um pequeno número de espécies, que são excessivamente exploradas, pondo em risco a sustentabilidade da floresta. São frequentes os problemas com a nomenclatura das madeiras, principalmente no que concerne às madeiras tropicais, dada a multiplicidade de designações comerciais existentes no mercado. Desta forma, uma mesma designação comercial é, em muitos casos, atribuída a madeiras muito diferentes, sem qualquer afinidade botânica. Adicionalmente, o fato de existirem diferentes dialetos nas circunscrições da floresta amazônica (consequência das suas características geográficas, posto que, a amazônia faz fronteira com Venezuela, Colômbia, Peru e fica muito próxima da Bolívia - vide Figura 1.1) contribui para aumentar a confusão alusiva à nomenclatura das madeiras, pois, indivíduos com descendências distintas costumam referenciar uma mesma espécie de árvore de maneira contraditória (Bessa, 2009, p. 2-4).

O desmatamento e seus impactos associados não são as únicas ameaças à integridade das florestas da Amazônia. A partir da década de 1990 extensas áreas de floresta têm sido anualmente empobrecidas pela degradação causada pela atividade madeireira. A exploração madeireira, se praticada com técnicas de manejo florestal, pode ser uma grande aliada no combate ao desmatamento. A atividade também contribui fortemente com a geração de empregos e renda, sobretudo, se praticada de forma leviana, a atividade madeireira passa a ser considerada um grande problema ambiental (JÚNIOR, 2007, p. 6987).

O primeiro órgão a oferecer apoio nessa pesquisa foi a *Embrapa*, especificamente uma unidade que se situa na floresta amazônica, e, em função disto, tem uma demanda constante alusiva à categorização de madeiras. Além disso, em função de interesses comuns, os resultados desse trabalho podem interessar a órgãos como *IBAMA* e *Polícia Federal*.

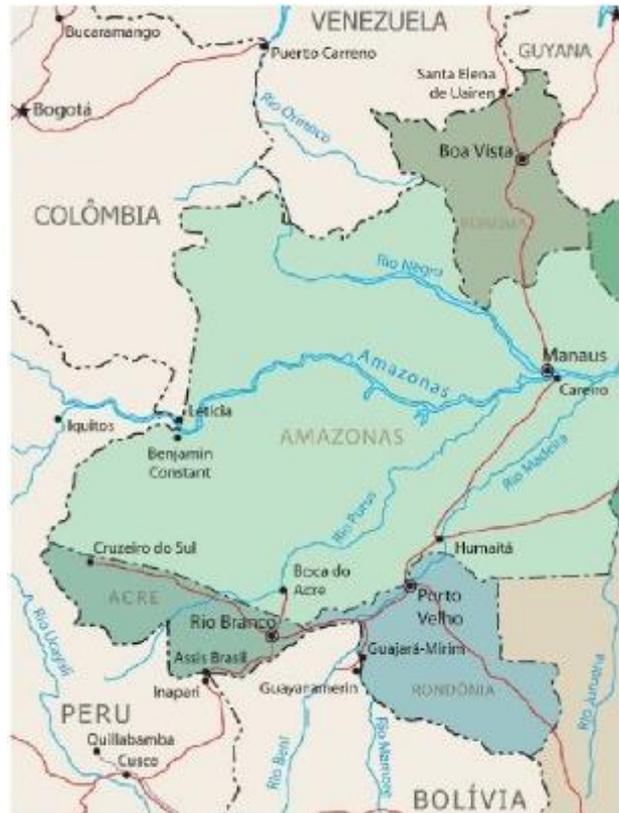


Figura 1.1: Mapa exibindo as fronteiras da Amazônia (Fonte: Fragmento da imagem vista em <http://www.brasil-turismo.com/mapas/mapa-politico.htm>)

1.1.1 Contextualização

A atividade madeireira legalizada (que utiliza técnicas de manejo florestal) não representa um problema, entretanto, a atividade madeireira clandestina é um dos motivos mais frequentes da degradação florestal, que leva à redução dos estoques de biomassa, à redução de espécies de valor comercial e a um ambiente propício às queimadas, que são responsáveis pelo aumento do risco de extinção de uma série de espécies (JÚNIOR, 2007, p. 6988).

Atualmente, o tráfico ilegal de madeira na Amazônia acontece, na maioria das vezes, da seguinte maneira: os fraudadores criam *notas frias*, ou seja, notas que especificam madeiras cuja extração é lícita, e que não correspondem, na realidade, às madeiras que estão sendo transportadas pelas alfândegas brasileiras. E os agentes alfandegários, por ignorância acerca da classificação da carga, são obrigados a liberar a passagem, conforme dados informais disponibilizados pela *Embrapa*.

Segundo Santos (2010, p. 23) existem várias especulações acerca das possíveis causas que expliquem as variações nas taxas anuais de desflorestamento. Partindo desses questionamentos, vários estudiosos deram início a inúmeros projetos. E esses projetos, baseados em análises espaciais e econométricas, afirmam que o desmatamento é o resultado de diferentes fatores, entre os quais é possível destacar:

- Os preços das *commodities* agropecuárias;
- Falta de infra-estrutura regional;
- Restrições geológicas para a expansão das atividades agropecuárias;
- Políticas públicas ineficientes para coibir e exploração ilegal de madeira;
- E, principalmente, o desmatamento clandestino e insustentável das florestas.

Segundo Santos (2010, p. 29), tanto no Brasil quanto no exterior, quando se fala em Amazônia muitos imaginam que ela está totalmente circunscrita dentro do Brasil, e isso não é verdade: A abrangência da floresta desconsidera a fronteira entre os países, e passa a ser referenciada como Pan-Amazônia, Amazônia Internacional ou Amazônia Continental. A Pan-Amazônia representa mais de 60% dos remanescentes de florestas tropicais do mundo, e é a maior área contínua de floresta em todo o globo. Aloca cerca de 25% de todas as espécies já identificadas de plantas e animais.



Figura 1.2: Mapa exibindo a Pan-Amazônia (Fonte: (SANTOS, 2010, p. 29))

Cerca de um quarto das florestas tropicais do mundo estão localizadas no Brasil. Toda essa “riqueza” está sendo ameaçada, principalmente, pelo desmatamento ilegal que se repete a cada ano (SANTOS, 2010, p. 29-36).

A Pan-Amazônia possui uma área de 6.943.912.63 Km², que está dividida da seguinte forma:

- Brasil (60,30%);
- Peru (11,31%);
- Bolívia (6,90%);
- Venezuela (6,74%);
- Guiana (3,03%);
- Suriname (2,09%);
- Equador (1,47%);
- Guiana Francesa (1,20%).

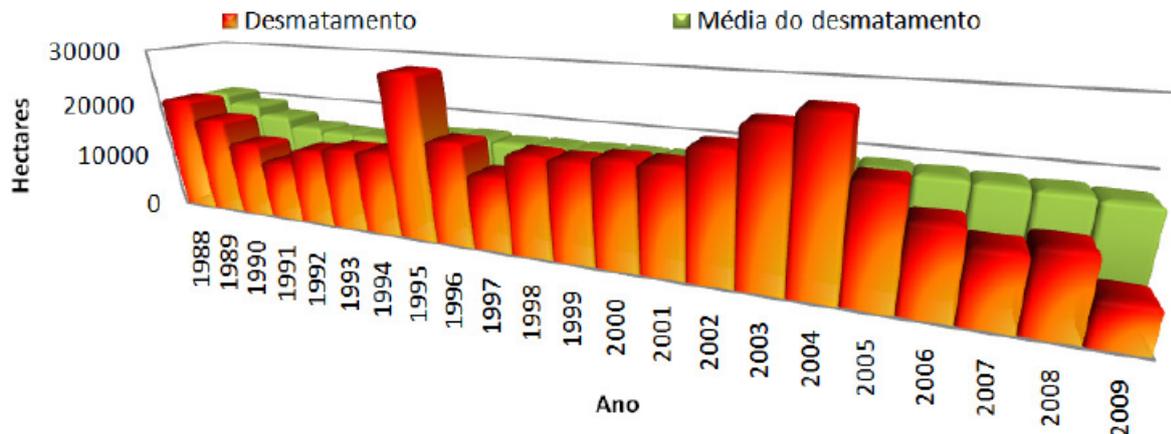


Figura 1.3: Gráfico do desmatamento (Fonte: (SANTOS, 2010, p. 40))

O desmatamento da Amazônia passa a ser enxergado como um problema ambiental de extremo risco para o ecossistema global, e tem efeitos diretos na conservação da biodiversidade, no clima, no ciclo de água e no controle de efeito estufa. O desmatamento também traz uma série de consequências sociais imediatas, pois, as famílias que se localizam nas imediações das regiões que foram diretamente afetadas pelo desmatamento, diversas vezes, acabam migrando para outras regiões sem as condições mínimas de higiene e conforto, e isso desencadeia uma série de consequências sociais (CARRERO, 2009, p. 1-13).

A Figura 1.3 mostra que as políticas públicas surtiram determinado efeito no que concerne à contenção do desmatamento, notadamente entre 2004 e 2009, sobretudo, considerando o prejuízo ecológico e social desencadeado por essa atividade, é necessário investir em formas mais eficientes de coibição do desmatamento ilegal das florestas, notadamente da Pan-Amazônia, vista na Figura 1.2.

Por fim, a constituição federal informa que:

“Art. 225. Todos tem direito ao meio ambiente ecologicamente equilibrado, bem de uso comum do povo e essencial à sadia qualidade de vida, impondo-se ao Poder Público e à coletividade o dever de defendê-lo e preservá-lo para as presentes e futuras gerações.

§ 1º Para assegurar a efetividade desse direito, incumbe ao Poder Público:

I - preservar e restaurar os processos ecológicos essenciais e prover o manejo ecológico das espécies e ecossistemas;

II - preservar a diversidade e a integridade do patrimônio genético do País e fiscalizar as entidades dedicadas à pesquisa e manipulação de material genético;

III - definir, em todas as Unidades da Federação, espaços territoriais e seus componentes a serem especialmente protegidos, sendo a alteração e a supressão permitidas somente através de lei, vedada qualquer utilização que comprometa a integridade dos atributos que justifiquem sua proteção;

IV - exigir, na forma da lei, para instalação de obra ou atividade potencialmente causadora de significativa degradação do meio ambiente, estudo prévio de impacto ambiental, a que se dará publicidade;”

(Fonte: CONSTITUIÇÃO DA REPÚBLICA FEDERATIVA DO BRASIL - Publicada no Diário Oficial da União nº 191-A, de 5 de outubro de 1988)

1.2 Objetivo

Até hoje o processo de classificação das madeiras, que circulam pelas alfândegas brasileiras, é feito pela observação, que geralmente é ineficiente, pois, os agentes alfandegários não possuem a capacitação necessária para classificar toda a diversidade de espécies traficadas (EMBRAPA, 1972). Assim, se a classificação for feita de maneira errada, a fiscalização fica impossibilitada de cumprir seu papel: impedir o tráfico de cargas ilícitas.

O objetivo deste trabalho é propor um modelo para automatizar do processo de classificação de madeiras tropicais, oriundas da Amazônia, especificamente de um escopo definido de madeiras, sobre o qual incidem os maiores índices de tráfico ilegal, nas alfândegas que fazem a circunscrição da floresta amazônica.

O produto final dessa dissertação poderá ser aperfeiçoado, em trabalhos futuros, para servir de ferramenta de apoio que, se utilizada pelas autoridades competentes, majorará a sustentabilidade dos recursos naturais da floresta amazônica, se tornando uma forte aliada para a economia da exploração legalizada das madeiras, que é sustentável em virtude da obrigatoriedade de técnicas de manejo florestal, segundo apontamentos de Júnior (2007, p. 6987).

1.3 Código fonte

Todo o código fonte, utilizado neste trabalho, está disponível na *Internet*. Para ter acesso ao referido código fonte, visite o site <http://alexpaulo.com>, abra a seção contato, e envie uma mensagem para o autor, informando seu e-mail, e explicando porque deseja utilizar o referido código. O autor responderá enviado um e-mail para o remetente desta mensagem, com o código solicitado em anexo.

1.4 Estrutura deste trabalho

Essa dissertação foi estruturada em cinco capítulos, um glossário e três apêndices, que disponibilizam o seguinte:

- **Capítulo um (INTRODUÇÃO):** Informa o que o governo já tem feito para combater o desmatamento ilegal de madeiras no Brasil. Descreve os objetivos deste trabalho. Informa como o código fonte deste trabalho pode ser adquirido. E, por fim, oferece uma visão geral acerca da estrutura geral deste manuscrito;
- **Capítulo dois (REVISÃO DE LITERATURA):** Esse capítulo traz um apanhado geral acerca dos assuntos que servem de alicerce para o trabalho desenvolvido:
 - Imagem digital (Seção 2.1.1);
 - Textura (Seção 2.1.2);
 - Visão computacional (Seção 2.1.3);
 - Etapas comuns no processamento digital de imagens (Seção 2.1.4);
 - Distribuição de frequência (Seção 2.1.5);
 - *Local Binary Patterns* (Seção 2.1.6);
 - Normalização (Seção 2.1.7);
 - *Instance Based Learning* (Seção 2.1.8);
 - *Cross-Validation* (Seção 2.1.9);
 - Ruído em imagens digitais (Seção 2.1.10).

Esse capítulo ainda cita algumas soluções alternativas para atenuar a tráfico ilegal de madeiras no Brasil. E faz um levantamento acerca de alguns trabalhos relacionados, pois, existem vários;

- **Capítulo três (MÉTODO PROPOSTO):** Esse capítulo descreve os algoritmos utilizados dentro de cada etapa do método proposto, e mostra como essas etapas se relacionam, ou seja, de que forma o resultado de uma serve de insumo para outra;
- **Capítulo quatro (EXPERIMENTOS):** Este capítulo aborda os resultados de cada experimento, informando as respectivas estatísticas e as inferências realizadas. Também é abordada a construção do banco de dados das imagens. As ferramentas utilizadas serão descritas. E, após a descrição do resultado de cada experimento, é realizada uma análise sobre a normalização, as técnicas de Cross-Validation e as métricas de distância;
- **Capítulo cinco (CONCLUSÕES):** Este capítulo apresenta conclusões acerca de todas as inferências desta pesquisa, e pode servir de subsídio para pesquisas correlacionadas, ou para a continuidade desta pesquisa;
- **Capítulo seis (TRABALHOS FUTUROS):** Este capítulo apresenta sugestões para a continuidade deste trabalho, levando em conta a experiência adquirida;
- **Glossário do manuscrito (GLOSSÁRIO):** Esta seção tem o objetivo de facilitar a leitura de pessoas que não possuam familiaridade com os termos técnicos utilizados neste trabalho;
- **Apêndice A (RESULTADOS DA QUINTA BATERIA DE TESTES):** Os resultados dos experimentos desta bateria de testes, disponíveis na Seção 4.2.5, ficaram extensos, e, por isso, pareceram mais adequados para um apêndice.

CAPÍTULO 2

REVISÃO DE LITERATURA

2.1 Referencial teórico

O objetivo desta Seção é abordar uma série de questões que servem de base estrutural para o que está dissertação se propõe a desenvolver.

2.1.1 Imagem digital

Os insumos utilizados para atingir os objetivos deste trabalho são uma série de imagens fornecidas pela *Embrapa* e pelo *VRI(UFPR)*. Desta forma, será realizada uma descrição formal acerca destes insumos.

Alguns autores definem uma imagem digital da seguinte forma:

“Uma imagem pode ser definida como uma **função bidimensional, $f(x, y)$** , em que **x** e **y** são coordenadas espaciais (plano), e a amplitude de **f** em qualquer par de coordenadas (x, y) é chamada de **intensidade ou nível de cinza** da imagem nesse ponto. Quando **x, y** e os valores de intensidade de **f** são quantidades finitas e discretas, chamamos a referida imagem de **imagem digital**.”

(Fonte: (GONZALEZ; WOODS, 2010, p. 1))

Ainda segundo Gonzalez e Woods (2010, p. 1), conforme se verifica no capítulo introdutório do seu livro, é possível afirmar que: “[...] uma imagem digital é composta de um número finito de elementos, cada um com localização e valor específicos. Esses elementos são chamados de **elementos pictóricos, elementos de imagem, pels e pixels**”.

Pixel foi o termo mais observado, se comparado com os sinônimos acima, em toda a revisão bibliográfica realizada durante este trabalho.

Segundo Jain, Kasturi e Schunck (1995, p. 12) um pixel é uma amostra da intensidade da imagem quantizada para um valor inteiro, e esse valor inteiro deve ser visto como uma

representação numérica para um fragmento da imagem. Esses autores também explicam que uma imagem é um vetor de pixels de duas dimensões. Os índices de linha e coluna $[i, j]$ de um pixel são valores inteiros que especificam a posição do pixel no referido vetor, assim, a referência $[0, 0]$ indica o pixel alocado no canto superior esquerdo da imagem.

Pelo estudo realizado, foi possível inferir que uma imagem digital é composta por um conjunto de pixels, e representa uma informação inteligível, que pode ser discernida/interpretada por aqueles que possuem conhecimento prévio alusivo à informação nela gravada.

Gonzalez e Woods (2010, p. 1) também explicam que quando se fala em processamento digital de imagens deve ser entendido que esse processamento necessariamente é o processamento de uma imagem digital por um computador digital.

2.1.2 Textura



Figura 2.1: Foto macro de uma mosca (Fonte: <http://www.cursodefotografia-gratis.com>)

Segundo Maenpaa (2003, p. 15), de maneira geral, uma textura pode ser definida como sendo um conjunto de características visuais e tácteis que são responsáveis pela aparência de algo. Texturas podem ser formadas por elementos muito pequenos, como grãos de areia, ou enormes, como as estrelas observadas em uma Seção da galáxia. Texturas também podem ser formadas, em determinada superfície, em função de variações na forma, na iluminação, nas sombras, nos reflexos, e em outras coisas.

Também é importante entender que a textura que se percebe é uma função da distância do observador: poucas pessoas sabem que os

olhos de uma mosca possuem a textura vista na Figura 2.1.

Segundo Pietikäinen et al. (2011, p. 3) a análise de textura tem sido um tópico intensamente pesquisado, desde a década 60, e uma grande variedade de técnicas para discriminar texturas foram propostas.

A maioria dos métodos propostos, ao contrário do que se esperava, não foram suficientemente capazes de descrever, de forma satisfatória, texturas do mundo real. E se mostraram muito complexos, do ponto de vista computacional, para atingir os pré-requisitos de processamento em tempo real, exigidos por muitas aplicações visuais.

Nesses últimos anos, muitos descritores de textura local (eficientes discriminativamente e computacionalmente) foram propostos, entre eles, é válido citar o *LBP* (abordado na Seção 2.1.6), que tem mostrado progressos significantes, no trato com as texturas, para vários problemas de visão computacional (PIETIKÄINEN et al., 2011, p. 4).

De uma maneira geral, a textura pode ser vista como um conjunto de variações de intensidade nos *pixels*. Essas variações formam certos padrões repetitivos, e, em algumas áreas da visão computacional (abordada na Seção 2.1.3), as texturas podem representar um desafio para trabalhos que envolvem segmentação (tratada na Seção 2.1.4) e/ou reconhecimento (apresentado na Seção 2.1.4) (VIEIRA et al., 2012, p. 1).

Ainda segundo Vieira et al. (2012, p. 1), além do *LBP* (visto na Seção 2.1.6), existem vários detectores de textura, entre os quais, ele descreve brevemente:

- *GLCM (Gray Level Cooccurrence Matrix)*: Utiliza matrizes de co-ocorrência com o intuito de caracterizar as propriedades de textura de uma imagem. Essas matrizes são construídas considerando os oito vizinhos de um pixel. Depois de identificar os vizinhos, quatorze medidas são consideradas, entre elas, contraste, correlação, variância e entropia;
- *SDH (Sum and Difference Histograms)*: Foi proposto objetivando reduzir o custo de armazenamento da matriz de co-ocorrência (*GLCM*), trocando-a por um histograma de somas e diferenças entre os *pixels*;
- *FLBP (Fuzzy Local Binary Pattern)*: estende a abordagem *LBP*, incorporando a lógica *fuzzy* na representação de padrões locais de textura. Resultados práticos, do trabalho deste autor, mostram que o *FLBP* produz algumas melhorias em relação aos resultados do *LBP*, entretanto, acabam desencadeando um custo computacional muito maior que o tradicional *LBP*, que é bem leve, conforme explanado na Seção 2.1.6;
- *SIFT (Scale Invariant Feature Transform)*: É formado pelo detector de pontos chaves ('*Key features*'), com base no cálculo da diferença de Gaussianas, e, também, pelo descritor que utiliza histogramas de gradientes orientados para descrever a vizinhança local dos pontos de interesse;

- *LFP (Local Fuzzy Pattern)*: Considera uma vizinhança local dos pixels da imagem, e, baseado na teoria de números *fuzzy*, calcula-se o *membership degree* (grau de pertinência) do *pixel* à região. Neste contexto, a intensidade dos tons de cinza que compõem a imagem são interpretados como números de um conjunto *fuzzy*.

Cada detector de textura, referenciado acima, foi abordado de forma muito superficial. Para mais informações, acerca de qualquer um deles, é recomendável a leitura do material escrito por Vieira et al. (2012), encontrado nas referências bibliográficas deste trabalho.

2.1.3 Visão computacional

A visão permite aos humanos perceber e entender o mundo que os rodeia. A visão computacional permite duplicar alguns efeitos da visão humana: viabiliza a percepção, e o entendimento, de uma imagem, através de recursos eletrônicos (SONKA; HLAVAC; BOYLE, 2007, p. 1).

“Não existe um acordo geral entre os autores em relação ao ponto em que o processamento de imagens termina, e outras áreas relacionadas, como a análise de imagens e a visão computacional, começam“ (GONZALEZ; WOODS, 2010, p. 1).

Segundo os autores Sonka, Hlavac e Boyle (2007, p. 1-3), fazer com que o computador passe a ter habilidade de “enxergar” não é tarefa fácil: o mundo visual dos humanos possui três dimensões (o cérebro humano visualiza imagens 3D), e quando o computador tenta analisar objetos num espaço 3D, os sensores visuais disponíveis (câmeras fotográficas, câmeras de vídeo, etc) quase sempre produzem imagens em duas dimensões (2D). E essa projeção para uma quantidade menor de dimensões resulta em uma perda enorme de informações. E os equipamentos que já produzem imagens em três dimensões retornam dados que muitas vezes são complexos de analisar. De tal forma que, à depender do objetivo da análise, às vezes se torna mais apropriado trabalhar com imagens em duas dimensões.

Essa “habilidade de enxergar”, conferida ao computador, pode ser referenciada como visão computacional. E esse é o alicerce do trabalho desenvolvido nesta dissertação.

Segundo Gonzalez e Woods (2010, p. 2), um paradigma útil seria considerar a abstração através da qual existem três tipos de processos computacionais relacionado ao trato de imagens:

- (a) **Processos de nível baixo**: Envolvem pré-processamento de imagens, objetivando, por

exemplo, diminuir ruído, realçar contrastes ou aguçar imagens. Em um processo de nível baixo tanto a entrada quanto a saída são imagens;

- (b) **Processos de nível médio:** Dizem respeito a tarefas de outra natureza, como (detecção de segmentos específicos da imagem) e classificação (reconhecimento) de objetos. Em um processo de nível médio as entradas, geralmente, são imagens, e as saídas são atributos extraídos de tais imagens (como bordas, contornos, identificação etc);
- (c) **Processos de nível alto:** São processos mais complexos que criam sentido para objetos identificados, geralmente estão relacionados com as funções cognitivas, normalmente associadas à visão.

Segundo Gonzalez e Woods (2010, p. 2-5), o processamento digital de imagens, atualmente, é muito importante para uma grande variedade de aplicações, como: a codificação do nível de intensidade em cores, para facilitar a interpretação de imagens radiográficas, e outras imagens utilizadas na indústria, na medicina e nas ciências biológicas; Constitui prática comum, entre os geólogos, a utilização das mesmas técnicas, ou de técnicas similares, para estudar padrões de poluição, a partir de imagens aéreas de satélite; Também já está bem disseminado que, no escopo da arqueologia, métodos de processamento de imagens têm restaurado satisfatoriamente imagens fotográficas borradas. Adicionalmente, entre os mais variados problemas de percepção automatizada, baseada em visão computacional, é possível citar:

- Reconhecimento automático de caracteres;
- Inspeção e montagem de produtos na indústria;
- Reconhecimento militar, para diversos fins;
- Identificação biométrica através de impressões digitais;
- Identificação biométrica através de íris;
- Identificação de resultados clínicos usando imagens radiográficas e amostras de sangue;
- Tratamento de imagens aéreas, e de satélites, para previsão do tempo e avaliação ambiental.

Ainda segundo os mesmos autores (Gonzalez e Woods (2010, p. 5-12)), as áreas de aplicação do processamento digital de imagens são tão variadas que acaba sendo conveniente

categorizá-las de alguma forma. Uma maneira corriqueira de categorizar as imagens é tomar como referência a sua fonte:

- **Imagens formadas por raios gama:** São utilizadas, principalmente, na medicina nuclear e nas observações astronômicas;
- **Imagens formadas por raios X:** Os raios X representam uma das fontes mais antigas de radiação EM (Eletromagnética) utilizadas para a formação de imagens. O diagnóstico médico é uma das utilizações mais conhecida dos raios X. Eles também são muito utilizados em áreas como astronomia e indústria;
- **Imagens na banda ultravioleta:** As aplicações de imagens desta natureza também são referenciadas como aplicações de “luz” ultravioleta. Essas aplicações são utilizadas para litografia, inspeção industrial, microscopia, lasers, imagens biológicas e observações astronômicas;
- **Imagens na banda visível e na banda infravermelha:** Como a humanidade interage diariamente com imagens dessa categoria, já era de se esperar que a criação destas imagens, para as mais diversas aplicações, aconteça com muito mais frequência que a criação de todas as outras. Essas bandas são utilizadas, comumente, de forma conjunta, em aplicações alusivas à microscopia ótica, à astronomia, ao sensoriamento remoto, à indústria e ao policiamento;
- **Imagens na banda de micro-ondas:** As imagens desta banda são utilizadas, principalmente, para viabilizar o radar. O radar é importante para muitas aplicações, pois, possibilita coletar dados em praticamente qualquer região a qualquer momento, independentemente de características como clima, condições de iluminação e ambiente;
- **Imagens na banda de rádio:** As principais aplicações das imagens obtidas na banda de rádio podem ser vistas na medicina e na astronomia. Na medicina, por exemplo, ondas de rádio são utilizadas em imagens por ressonância magnética - MRI (*Magnetic Resonance Imaging*);
- **Outros exemplos de imagem:** Mesmo considerando que as imagens no espectro eletromagnético são muito mais utilizadas que as imagens nos outros espectros, existem várias outras modalidades que também são relevantes, tais como: a imagem acústica, a microscopia eletrônica e a imagem sintética (gerada por computador).

Segundo os autores Sonka, Hlavac e Boyle (2007, p. 3-5), existem vários fatores que contribuem para tornar a visão computacional um processo complexo:

- **A perda de informação quando se interpreta um mundo 3D à partir de imagens 2D:** Essa perda de informação acontece quando se captura uma imagem à partir de dispositivos eletrônicos que foram concebidos para retornar imagens em duas dimensões (é o que acontece com a expressiva maioria dos dispositivos concebidos para capturar imagens);
- **Dificuldade para identificar o tamanho real dos objetos:** Um pequeno objeto, que esteja perto de uma câmera fotográfica, pode “ficar do mesmo tamanho” de um objeto enorme, que esteja muito distante da câmera. E o observador da imagem precisará de conhecimento prévio, acerca desses dois tipos de objetos, para interpretar corretamente a imagem capturada. Se o referido conhecimento não existir, a imagem será interpretada de forma equivocada;
- **A interpretação exige conhecimento prévio:** A interpretação de uma imagem constitui a principal ferramenta da computação visual para abordar problemas que os humanos resolvem, muitas vezes, até sem entender como. Quando um humano tenta entender uma imagem, o cérebro faz uma busca por conhecimentos prévios, que são frutos de suas experiências, acerca do significado da imagem em análise. As diversas pesquisas relacionadas às novas técnicas de inteligência artificial, desenvolvidas ao longo das últimas décadas, já resultaram em avanços significativos, entretanto, atualmente, a habilidade que pode ser observada em uma máquina, devidamente programada para “entender imagens”, ainda é muito limitada;
- **Ruído:** Todas as informações do mundo real estão naturalmente acompanhadas de algum nível ruído. Existem ferramentas matemáticas concebidas com o objetivo de tratar esses ruídos. Obviamente, a utilização de ferramentas dessa natureza torna o processo de análise de imagens mais custoso do ponto de vista computacional;
- **Muitos dados para serem analisados:** Imagens e sequências de vídeo podem ser enormes. Se o processamento que precisa ser realizado com eles não for simples, então, pode se tornar muito difícil atingir uma performance em tempo real, o que significa, no escopo do processamento de um vídeo, tratar entre 25 e 30 imagens por segundo;
- **Influência dos efeitos físicos naturais:** Uma imagem pode ser afetada por vários processos físicos naturais, como, a luz do ambiente, a posição do observador (ou do dispositivo

eletrônico que capturará as imagens), a geometria local, e as propriedades de reflexão, da superfície sobre a qual o objeto em análise está posicionado. Por tais razões, é preciso evitar esses efeitos durante a captura das imagens, pois, os objetivos finais do processamento dependem da interpretação das imagens capturadas;

- **Visão à partir de um buraco de chave:** Quando o cérebro humano analisa uma imagem, quase sempre ele considera um universo de elementos que formam a vizinhança da imagem analisada. Isso seria equivalente a considerar o conteúdo de um livro no qual um parágrafo específico está sendo analisado. Uma análise bem contextualizada tem maiores possibilidades de estar correta. Geralmente o computador analisa imagens "visualizadas à partir de um buraco de chave", ou seja, trabalha com "recortes" de uma imagem muito maior. Imagine um Quadro com uma imagem que lhe seja familiar, cole nesse Quadro algumas moedas, em seguida pinte todo o Quadro com tinta preta, por fim, remova as moedas e analise, separadamente, cada recorte de imagem que foi preservado sem tinta, em baixo de cada moeda, quase sempre o computador é obrigado a analisar pequenos recortes como esses da metáfora utilizada, desconsiderando todo um contexto de informações que poderiam auxiliar na interpretação. Assim, é possível perceber que a visão computacional, desde a etapa de aquisição de imagem (abordada na Seção 2.1.4), sofre várias limitações.

2.1.4 Etapas comuns no processamento digital de imagens

Segundo Gonzalez e Woods (2010, p. 15-18) existem várias abordagens que podem ser utilizadas no processamento digital de imagens, tais como:

- **Aquisição:** Referenciando de forma resumida, a aquisição é o processo de captura de uma imagem, entretanto, esse processo pode englobar alguns passos de pré-processamento adicionais, como o redimensionamento da imagem;
- **Realce:** É o fato de manipular uma imagem para que o resultado da manipulação seja mais útil para um processo específico, segundo a subjetividade de público que trabalhará com a imagem. Uma técnica de realce utilizada para radiografias pode, perfeitamente, não ser adequada para realçar imagens de satélite. Assim, a técnica indicada depende do problema específico;
- **Restauração:** Geralmente se baseia em modelos matemáticos ou probabilísticos de de-

gradação de imagens, constituindo um processo totalmente objetivo, ao contrário do realce, que se baseia na subjetividade humana acerca do que constitui uma "boa" imagem;

- **Compressão:** Como o próprio nome já informa, trata de técnicas destinadas a reduzir o tamanho dos arquivos que representam as imagens, e isso majora a performance de SGBDs/redes que precisem armazenar/trafegar tais imagens;
- **Transmissão:** Como qualquer informação digital, a imagem digital pode ser transmitida em uma rede de computadores, seja uma rede com fio ou uma rede *wireless*, em uma LAN, MAN ou WAN;
- **Segmentação:** É referenciada como uma das tarefas mais difíceis no escopo do processamento digital de imagens, e possui o objetivo de capturar recortes específicos, identificados de acordo com alguma lógica considerada, por exemplo, para a análise biométrica da imagem de um olho humano, interessa, apenas, a imagem correspondente à íris, dessa forma, seria necessário desconsiderar todo o resto da imagem (cílios, pupila, esclerótica e a pele da face), a Figura 2.2 ajuda a entender essa assertiva;

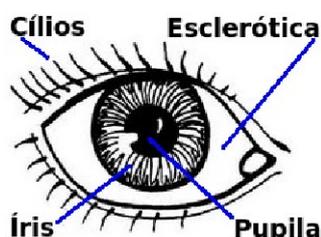


Figura 2.2: Olho humano

- **Classificação/Reconhecimento:** É o processo de identificar/classificar uma imagem com base em um conhecimento prévio. Muitas vezes é executado após o processo de segmentação. Atribui um rótulo à imagem analisada ("Círculo", "Retângulo", "Losângulo", etc). Ainda fazendo analogia à Figura 2.2, o rótulo atribuído a uma íris, devidamente segmentada, poderia ser "José", "Maria" ou, preferencialmente, um valor que identificasse de maneira única um indivíduo (CPF, RG, e-mail, ou qualquer valor que fosse exclusivo por definição).

Segundo os autores Sonka, Hlavac e Boyle (2007, p. 7) se espera que a visão computacional resolva tarefas muito complexas, considerando que a meta é obter resultados similares àqueles

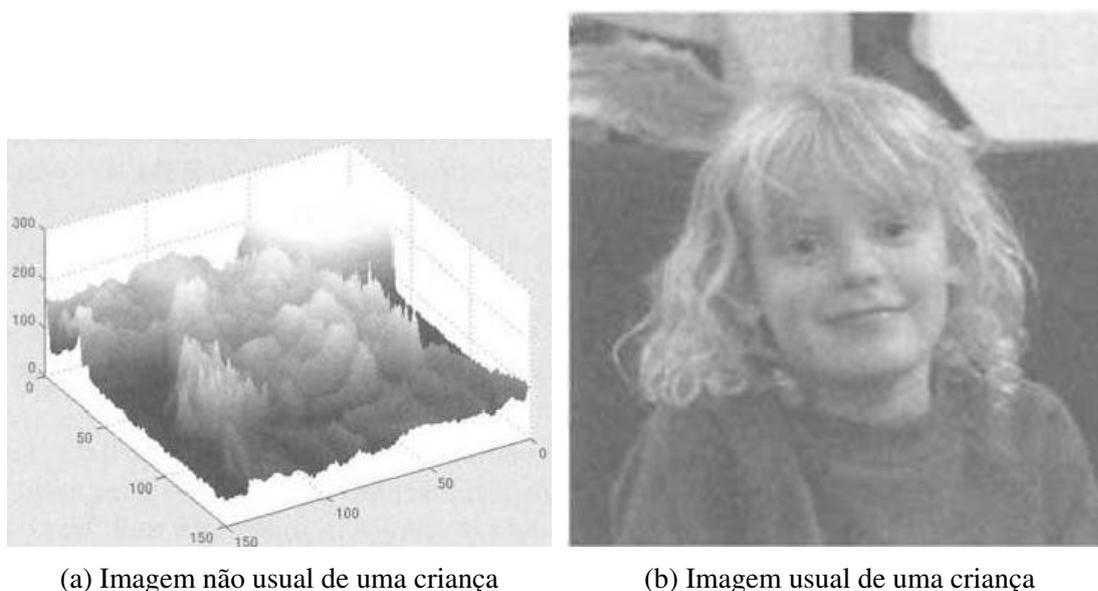


Figura 2.3: Imagens de uma criança (Fonte: (SONKA; HLAVAC; BOYLE, 2007, p. 7-8))

fornecidos pelos sistemas biológicos. Objetivando evidenciar a complexidade dessa tarefa, observe as Figura 2.3 (a) e (b).

Certamente não parece, entretanto, as Figuras 2.3a e 2.3b são formas distintas de exibir a mesma imagem: o rosto de uma criança.

Na Figura 2.3a o eixo vertical informa o brilho correspondente ao pixel localizado segundo as coordenadas dos outros dois eixos. Na Figura 2.3b se vê a mesma imagem, agora representada de uma forma inteligível para o cérebro humano. Mesmo à despeito de ambas as representações possuírem exatamente a mesma informação, para um observador humano, é difícil identificar uma correlação entre essas duas formas de representação, e, sem observar a Figura 2.3b, é improvável que alguém consiga discernir que se trata do rosto de uma criança.

O cerne da questão é: o cérebro humano utiliza uma série de conhecimentos prévios para interpretar/reconhecer qualquer imagem que seja apresentada a ele, enquanto o computador apenas dispõe de um vetor com uma série de valores inteiros para realizar o mesmo trabalho.

Os valores disponibilizados para o computador, sob o prisma do cérebro humano, seriam visualizados de forma similar ao que se vê na Figura 2.3a, dessa forma, é fácil perceber a complexidade envolvida nas tarefas de reconhecimento, no âmbito do processamento digital de imagens.

2.1.5 Distribuição de frequência

Quando se trabalha com grandes conjuntos de dados é conveniente agrupar tais dados em classes, categorias, intervalos ou subconjuntos. O objetivo desse agrupamento é entender melhor o todo. Esse agrupamento é referenciado como *Distribuição de Frequência* ou simplesmente como *Distribuição* (FREUND; SIMON, 2000, p. 25).

Tabela 2.1: Exemplo de distribuição de frequência numérica por faixa

Peso (gramas)	Número de Espécimes
0.0 - 19.9	16
20.0 - 39.9	38
40.0 - 59.9	35
60.0 - 79.9	20
80.0 - 99.9	11
100.0 - 119.9	4
120.0 - 139.9	1
Total 125	

Fonte: (FREUND; SIMON, 2000, p. 29)

Quando os dados se apresentam agrupados de acordo com valores numéricos, eles são referenciados como distribuição numérica ou distribuição quantitativa. Nessas distribuições, cada classe pode ser representada por uma faixa de valores (como no caso da Tabela 2.1 - que exhibe os pesos de um conjunto de animais apreendidos) ou pode ser representada por um único valor (como no caso da Tabela 2.2 - que relata as visitas, de um grupo específico de pessoas, ao zoológico) (FREUND; SIMON, 2000, p. 25)

Tabela 2.2: Exemplo de distribuição de frequência numérica por valor

Nº visitas ao Zôo	Número de pessoas
0	90
1	72
2	26
3	8
4	3
5	0
6	1
Total 200	

Fonte: (FREUND; SIMON, 2000, p. 25)

Quando os dados estão agrupados em categorias não numéricas, eles são referenciados como distribuição por categoria ou distribuição qualitativa (como no caso da Tabela 2.3 - que exhibe reclamações dos usuários das empresas aéreas) (FREUND; SIMON, 2000, p. 26).

Tabela 2.3: Exemplo de distribuição de frequência por categoria

Natureza da Reclamação	Número de Reclamações
Espaço insuficiente para as pernas	719
Assentos desconfortáveis	914
Corredores estreitos	146
Espaço insuficiente para bagagem de mão	218
Banheiros insuficientes	58
Outras queixas	384
	Total 2439

Fonte: (FREUND; SIMON, 2000, p. 26)

Segundo Freund e Simon (2000, p. 26), também é possível transformar uma distribuição qualitativa em uma distribuição numérica, representando cada categoria por um número. A construção de uma Tabela de frequência é feita com base em três passos:

- Escolha das classes (intervalos);
- Enquadramento dos dados brutos nessas classes;
- Contagem do número de elementos em cada classe da distribuição.

Ainda segundo Freund e Simon (2000, p. 26), o objetivo principal de uma distribuição de frequência é apresentar os dados de forma relativamente compacta, dando uma boa visualização global do todo.

Histogramas

Para as distribuições de frequência, a forma gráfica mais comum de representação é o histograma, como o que pode ser visto na Figura 2.4. Em um histograma, o eixo vertical representa as frequências que podem ser associadas a cada classe, e o eixo horizontal representa, para cada barra do histograma, uma classe cuja frequência será mensurada em razão diretamente proporcional à altura da barra considerada (FREUND; SIMON, 2000, p. 31).

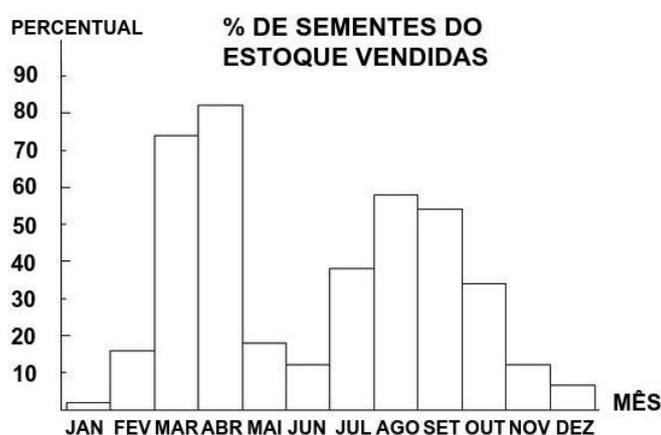


Figura 2.4: Histograma

Distribuições de frequência referentes a dados, armazenados em *SGBDs* (*Sistemas Gerenciadores de Bancos de Dados*), costumam ser bastante úteis, pois, permitem responder uma série de questões de forma mais ágil, posto que, as buscas são realizadas considerando os dados exibidos pelo histograma, que já estão sumarizados (LOANNIDIS, 2003, p. 2).

Ainda segundo Freund e Simon (2000, p. 31), é importante notar que os retângulos de um histograma vão de uma fronteira de classe até a fronteira seguinte: o mesmo ponto que serve de término para uma classe representa o início da outra, de maneira que, não se enxergam "brechas" entre os retângulos de um histograma, conforme pode ser observado na Figura 2.4.

Pelo que já foi exposto, é possível perceber que um histograma é uma representação gráfica útil para exibir os dados de uma distribuição de frequência. E, pelas pesquisas realizadas neste trabalho, também foi possível perceber que se trata de uma ferramenta comum, nos trabalhos que envolvem processamento digital de imagens.

Histograma de uma imagem

Segundo Teo (2003, p. 26), o histograma de uma imagem é um gráfico que representa a distribuição de intensidades, ou níveis de cinza, da referida imagem. O histograma de uma imagem digital com níveis de cinza na faixa $[0, L - 1]$ pode ser representado pela Equação 2.1:

$$p(r_k) = \frac{n_k}{n} \quad (2.1)$$

Para esta função discreta, r_k representa o k -ésimo nível de cinza; n_k representa o número de pixels da imagem que possui este k -ésimo nível de cinza; n é o número total de pixels da imagem; e, por fim, $k = 0, 1, 2, \dots, L - 1$. A Figura 2.5 exibe algumas imagens e seus respectivos histogramas.

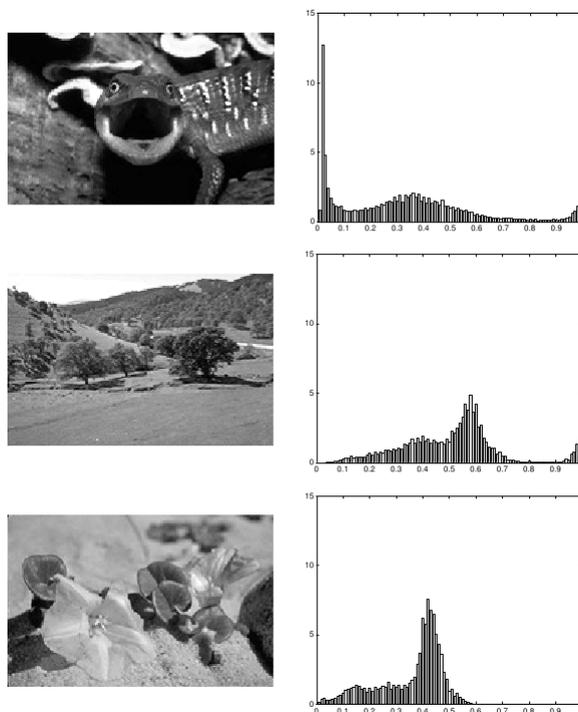


Figura 2.5: Histograma de intensidade de pixel (Fonte: Adaptado de (ODOM, 2006, p. 5))

A Figura 2.6 exibe os aspectos percebidos quando os histogramas são gerados à partir de imagens escuras (2.6a), claras (2.6b), com baixo contraste (2.6c) e com alto contraste (2.6d). De maneira que, já é possível ter uma idéia de algumas características da imagem apenas olhando para o seu histograma (TEO, 2003, p. 26).

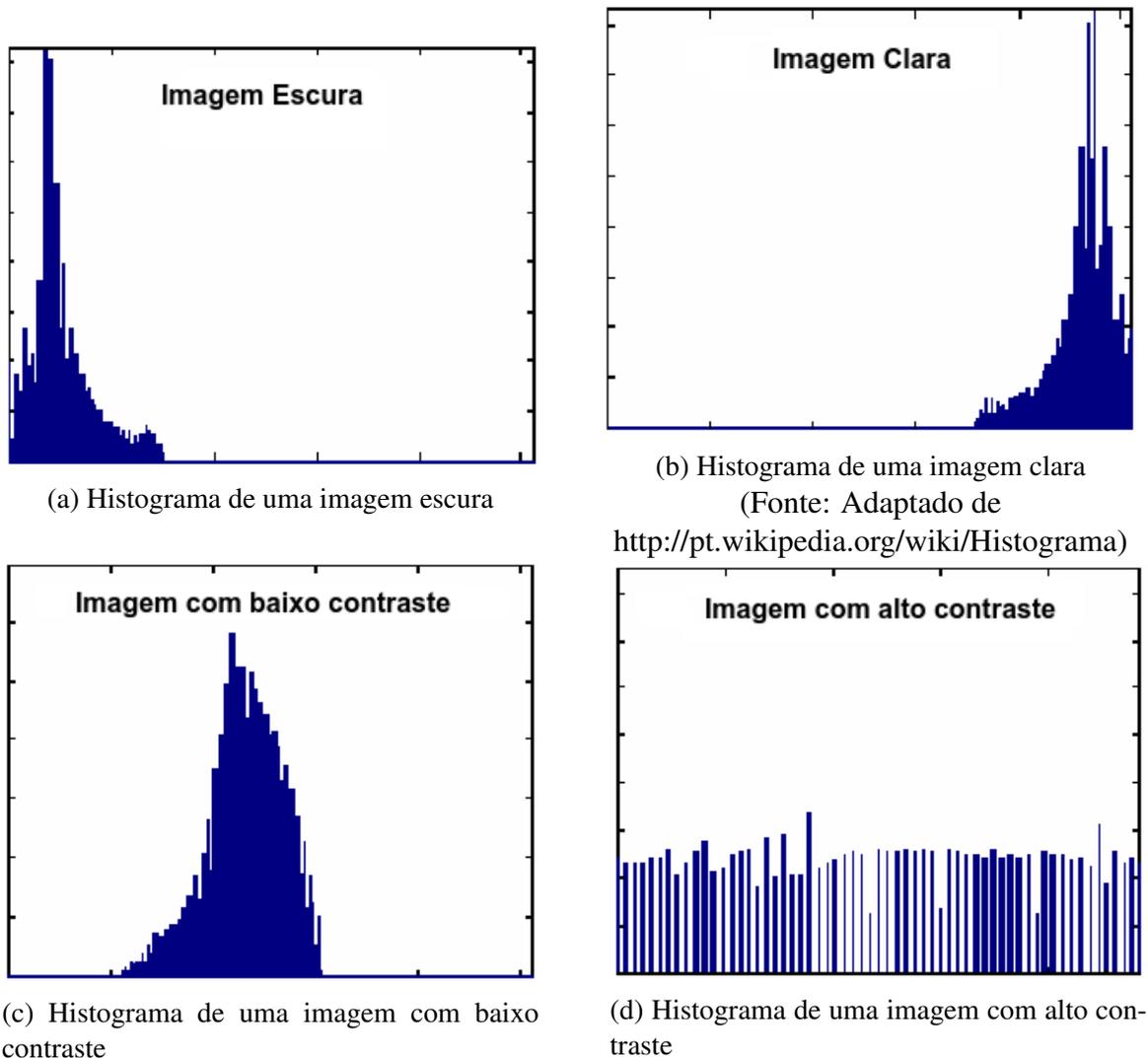


Figura 2.6: Histogramas (Fonte: Adaptado de (TEO, 2003, p. 26))

No que diz respeito ao processamento de imagens, os histogramas são utilizados como meio de visualização de distribuições de frequência referentes a determinadas características. Adicionalmente, comparar as distribuições de frequência de diferentes imagens é uma alternativa para julgar a similaridade entre elas (ZHONG, 2008, p. 22).

2.1.6 LBP - Local Binary Patterns

O *LBP* (*Local Binary Patterns*) é um operador para texturas considerado simples e eficiente. Ele se baseia numa idéia trivial: atribui um rótulo (representado por um número binário) a cada *pixel* analisado. Para atribuir esse rótulo, ele verifica se cada vizinho, do *pixel* em análise, em uma raio considerado, extrapola, ou não, um limiar (*threshold*) que é determinado pelo valor do *pixel* central, cujo rótulo está sendo determinado. Cada vizinho que extrapolar esse limiar (*threshold*), ou se igualar a ele, será marcado com o dígito binário 1, caso contrário, será marcado com o dígito binário 0. A junção desses dígitos binários representará o *pixel* central, lembrando que é importante seguir uma lógica para determinar qual o dígito mais significativo, ou, em outras palavras, à partir de que vizinho o número começará a ser formado, já que a vizinhança é circular. Também é preciso observar que quando se determina o dígito mais significativo, automaticamente está sendo atribuído um peso a cada um dos dígitos. A soma da multiplicação de cada dígito pelo seu respectivo peso retornará o que é referenciado como código *LBP* (PIETIKÄINEN et al., 2011, p. 4-5).

Ainda segundo Pietikäinen et al. (2011, p. 4), possivelmente, a propriedade mais importante do operador *LBP*, em aplicações do mundo real, é a sua invariância, percebida quando acontecem variações monotônicas nos níveis de cinza, que podem ser causadas por variações na iluminação do ambiente, no momento da captura da imagem. Outra característica igualmente importante é a sua simplicidade computacional, que torna possível enfrentar desafios nos quais as imagens precisam ser analisadas em tempo real. A Figura 2.7 mostra uma aplicação do *LBP*.

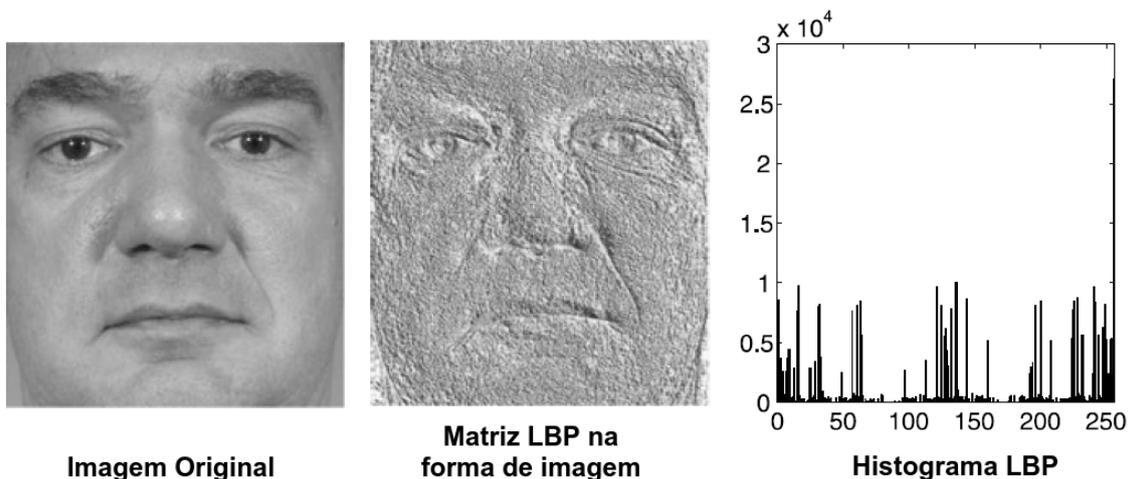
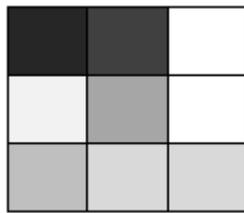


Figura 2.7: Aplicação do LBP (Fonte: Adaptado de (PIETIKÄINEN et al., 2011, p. 14))

Segundo Petrou e Sevilla (2006, p. 574) a idéia básica da abordagem *LBP* pode ser de-

monstrada na Figura 2.8, considerando uma vizinhança de 3×3 , todos os vizinhos que tiverem um valor igual ou maior que o *pixel* analisado (o *pixel* central) serão representados pelo valor 1, e todos os vizinhos que tiverem um valor inferior ao referido *pixel* serão representados pelo valor 0. O número binário formado por esses oito dígitos (considerados no sentido horário), ou o seu equivalente no sistema decimal, pode ser associado ao *pixel* central, e utilizado para caracterizar a textura local em análise.

É importante entender que, considerando o sentido horário, teremos oito números binários diferentes, dependendo daquele que for escolhido como dígito mais significativo. E, escolher o dígito mais significativo é o mesmo que associar, aos oito vizinhos do *pixel* central, pesos equivalentes a $2^7, 2^6, \dots, 2^0$: O *pixel* representado pelo dígito mais significativo teria o peso 2^7 em relação ao *pixel* central, e, analogamente, o *pixel* representado pelo dígito menos significativo teria o peso 2^0 em relação ao *pixel* central. Obviamente, alguns vizinhos serão mais importantes que outros, e, por conseguinte, é possível perceber que essa representação é sensível à rotação, ou, em outras palavras, é sensível à escolha do vizinho que representará o dígito mais significativo do número a ser formado (PETROU; SEVILLA, 2006, p. 574).



(a) Visão macro dos pixels

A	B	C
0	0	1
D	E	F
1		1
G	H	I
1	1	1

(b) Binários associados

Figura 2.8: Passos *LBP* (Fonte: Adaptado de (PETROU; SEVILLA, 2006, p. 574))

Ainda segundo Petrou e Sevilla (2006, p. 574) existe uma desvantagem nessa abordagem: ela detecta texturas muito pequenas, em função de considerar apenas os vizinhos imediatos (os *pixels* que estão adjacentes ao *pixel* analisado). Desta forma, a abordagem explanada aqui é apropriada apenas para detecção de micro-texturas, jamais para a detecção de texturas maiores.

Deixando o *LBP* invariante quanto à rotação

Para deixar o *LBP* invariante quanto à rotação, basta considerar todos os números que podem ser formados pela variação do dígito mais significativo, e escolher o menor deles. Como o exemplo estudado (Figura 2.8) considera oito vizinhos, selecionando cada um deles como sendo o

primeiro dígito mais significativo do número a ser formado, teremos oito números distintos:

- Iniciando em **A**: $00111111 = 63$;
- Iniciando em **B**: $01111110 = 126$;
- Iniciando em **C**: $11111100 = 252$;
- Iniciando em **F**: $11111001 = 249$;
- Iniciando em **I**: $11110011 = 243$;
- Iniciando em **H**: $11100111 = 231$;
- Iniciando em **G**: $11001111 = 207$;
- Iniciando em **D**: $10011111 = 159$.

Após considerar todas as formações possíveis, basta selecionar a menor. Nesse caso específico, o número selecionado, no escopo da abordagem que deixaria o *LBP* invariante quanto à rotação, seria 63 (00111111) (PETROU; SEVILLA, 2006, p. 574).

Tornando o LBP apropriado para macro-texturas

É possível tornar o *LBP* apropriado para macro-texturas aplicando a metodologia das vizinhanças circulares, através do incremento do raio a ser considerado. Essa abordagem resulta em uma representação de variadas resoluções para a textura analisada, uma vez que, para cada raio considerado, um padrão binário diferente será produzido. Neste contexto, considerando que o raio da vizinhança aumentasse, o número de vizinhos aumentaria exponencialmente, e isso desencadearia uma situação na qual os limites de processamento computacional rapidamente seriam alcançados, em função da necessidade de tratar números muito grandes. Objetivando evitar essas situações, as seguintes abordagens podem ser adotadas (PETROU; SEVILLA, 2006, p. 575):

- O algoritmo considerará apenas os pixels que estiverem exatamente dentro da linha formada pelo raio considerado, descartando, desta forma, os pixels que estiverem entre o pixel central (em análise) e os pixels que formam a referida linha imaginária;
- O algoritmo só considerará os pixels que estiverem a uma certa distância angular dos outros, conforme pode ser verificado na Figura 2.9b.

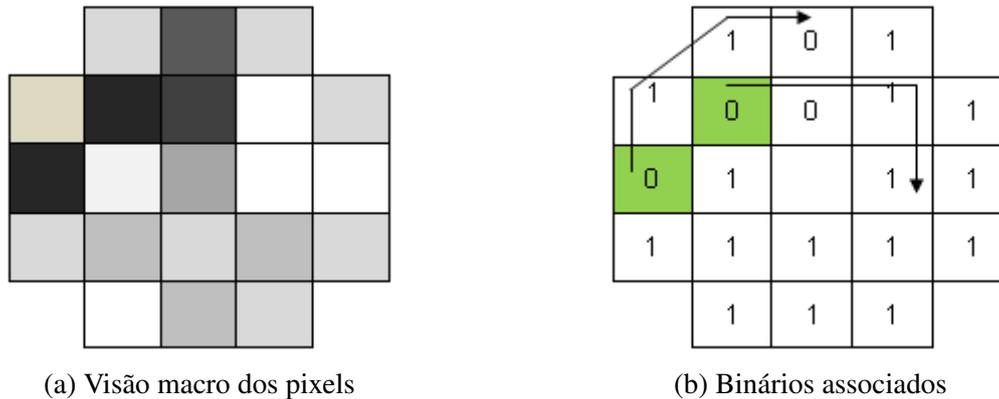


Figura 2.9: *LBP* para macro textura (Fonte: Adaptado de (PETROU; SEVILLA, 2006, p. 575))

A Figura 2.9b considera duas vizinhanças circulares, com raio 1 e raio 2, em torno do *pixel* central, e, à partir de cada vizinhança é formado o menor número binário possível, conforme explicado anteriormente. Considerando o exemplo específico, o menor número binário da vizinhança de raio 1 é 00111111 (63 em decimal), e o menor número binário para a vizinhança de raio 2 é 011011111111 (1791 em decimal). Em ambos os casos os números binários são construídos à partir da leitura dos vizinhos, em sentido horário, selecionando-se o menor binário formado. Para o exemplo em questão, os vizinhos selecionados como pontos iniciais estão marcados, na referida Figura, com a cor verde. Sobretudo, mesmo entendendo que apenas os *pixels* da "linha imaginária" do raio adotado serão considerados, à medida que o raio aumentar, o número de *pixels* a ser considerado aumentará muito rápido, dessa forma, uma maneira de atenuar as exigências de processamento computacional seria: à partir da vizinhança de raio 2, considerar os vizinhos alternadamente (o primeiro será considerado, o segundo não, o terceiro sim, o quarto não, e assim por diante...). Colocando essa abordagem em prática, para o exemplo específico, o menor binário encontrado na vizinhança de raio 2 foi 011011111111 (1791 em decimal), e esse valor seria representado por 011111 (31 em decimal) (PETROU; SEVILLA, 2006, p. 575).

Como calcular o contraste de uma vizinhança LBP

A mensuração de contraste (*contrast measure - C*) é obtida subtraindo a média dos níveis de cinza que possuem valor inferior ao nível de cinza do *pixel* central da média dos níveis de cinza que possuem valor igual ou superior ao nível de cinza do *pixel* central. Vale mencionar que se todos os valores, limiarizados em função do *pixel* central, forem iguais (todos iguais a 0 ou todos iguais a 1) isso denota ausência de contraste, nesta situação, o *C* em questão seria igual a

zero. A Equação 2.2 demonstra esse cálculo (PIETIKÄINEN et al., 2011, p. 4-5).

Exemplo	Limiarizado	Pesos																											
<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 5px;">6</td><td style="padding: 5px;">5</td><td style="padding: 5px;">2</td></tr> <tr><td style="padding: 5px;">7</td><td style="padding: 5px;">6</td><td style="padding: 5px;">1</td></tr> <tr><td style="padding: 5px;">9</td><td style="padding: 5px;">8</td><td style="padding: 5px;">7</td></tr> </table>	6	5	2	7	6	1	9	8	7	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td></tr> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;"></td><td style="padding: 5px;">0</td></tr> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">1</td><td style="padding: 5px;">1</td></tr> </table>	1	0	0	1		0	1	1	1	<table border="1" style="border-collapse: collapse; width: 60px; height: 60px;"> <tr><td style="padding: 5px;">1</td><td style="padding: 5px;">2</td><td style="padding: 5px;">4</td></tr> <tr><td style="padding: 5px;">128</td><td style="padding: 5px;"></td><td style="padding: 5px;">8</td></tr> <tr><td style="padding: 5px;">64</td><td style="padding: 5px;">32</td><td style="padding: 5px;">16</td></tr> </table>	1	2	4	128		8	64	32	16
6	5	2																											
7	6	1																											
9	8	7																											
1	0	0																											
1		0																											
1	1	1																											
1	2	4																											
128		8																											
64	32	16																											
Pattern = 11110001	LBP = 1 + 16 + 32 + 64 + 128 = 241																												

$$\text{Contraste} = \frac{6+7+9+8+7}{5} - \frac{5+2+1}{3} = 4,7 \quad (2.2)$$

O código *LBP* e seu cálculo de contraste (*LBP/C*) são frequentemente utilizados nos processos de segmentação e classificação (PIETIKÄINEN et al., 2011, p. 5) (para mais detalhes acerca de segmentação e classificação leia a Seção 2.1.4).

Aplicações de LBP no trato com madeira

Segundo Maenpaa (2003, p. 65) uma das aplicações do *LBP* relatadas com mais frequência é a inspeção de madeira. Uma abordagem comum, nessas aplicações, tem sido utilizar características de cor e textura combinadas com um método de *clustering* não supervisionado, como o *SOM* (*Self-Organizing Map*). Segundo Niskanen (2003, p. 70) o algoritmo *SOM* cria *clusters* por agrupar elementos com características semelhantes (referenciados como *BMU* - *Best Matching Unit*), implementando uma lógica similar à lógica implementada pelo algoritmo *Nearest Neighbor* (explicado na Seção 2.1.8), porém, com o intuito de agrupar os dados. O autor informa que a utilização do *LBP* melhorou a performance do sistema em 1%, entretanto, considerando que a indústria da madeira processa volumes enormes de informação, esse aumento é importantíssimo.

Às vezes é necessário simular situações do mundo real para mensurar, adequadamente, o poder de resposta dos algoritmos utilizados: No trabalho intitulado como *A visual training based approach to surface inspection* (NISKANEN, 2003, p. 105), que também faz uso do *LBP*, julgaram necessário organizar um ambiente que simulasse um tronco de madeira rolando em uma esteira de fábrica. Para atingir tais objetivos, construíram o ambiente que pode ser visto na Figura 2.10.

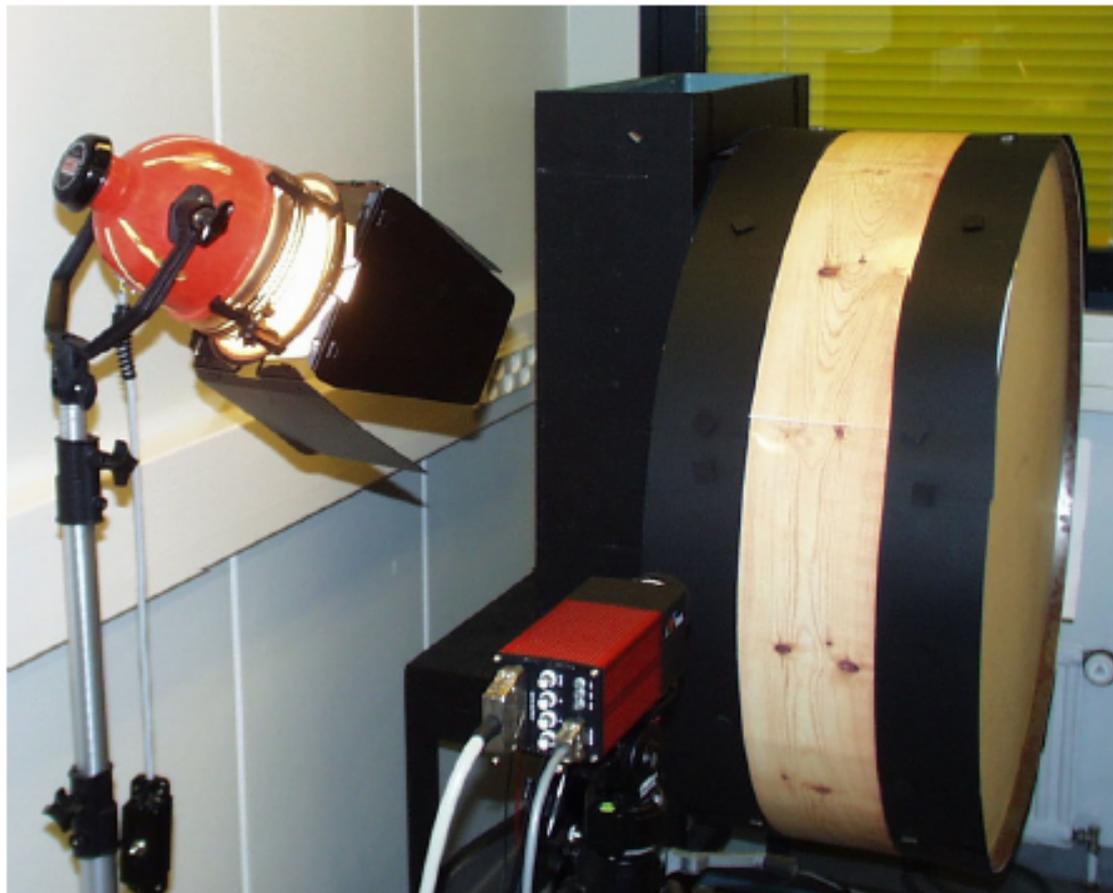


Figura 2.10: Simulando ambiente de inspeção de madeira (Fonte: (NISKANEN, 2003, p. 105))

Existem vários trabalhos que utilizam o *LBP* em aplicações que tratam imagens de madeira, conforme se vê no trabalho de (NISKANEN; SILVEN; KAUPPINEN, 2009, p. 2-7), intitulado como *Color and texture based wood inspection with non-supervised clustering*. Neste trabalho, o *LBP* representa um dos recursos utilizados para atingir os objetivos. Na conclusão do trabalho em questão foi encontrada assertiva: As abordagens *Centile* e *LBP* são baratas computacionalmente, e, combinadas, são eficientes na detecção de defeitos nas madeiras. Essa combinação representa uma base atrativa para uma metodologia de inspeção industrial.

Aplicações frequentes do LBP

LBP como descritor de faces Segundo Pietikäinen et al. (2011, p. 151-152), na abordagem *LBP* para classificação de texturas, as ocorrências dos códigos LBP eram alocados em um histograma. Daí por diante, as inferências eram estabelecidas pela verificação das simi-

laridades entre os histogramas. Surpreendentemente, considerar esta mesma abordagem para representação de imagens de face resulta em perda de informações espaciais, e, desta forma, para este tipo de aplicação, é necessário codificar as informações acerca das texturas retendo, também, informações acerca da localidade específica dos fragmentos de imagem analisados. Uma maneira de atingir esse objetivo é utilizar o descritor de textura *LBP* para construir vários descritores locais da face, e combinar tais descritores em um único descritor global, através da concatenação das informações dos descritores locais. Observar a Figura 2.11 ajudará a entender essas afirmações.

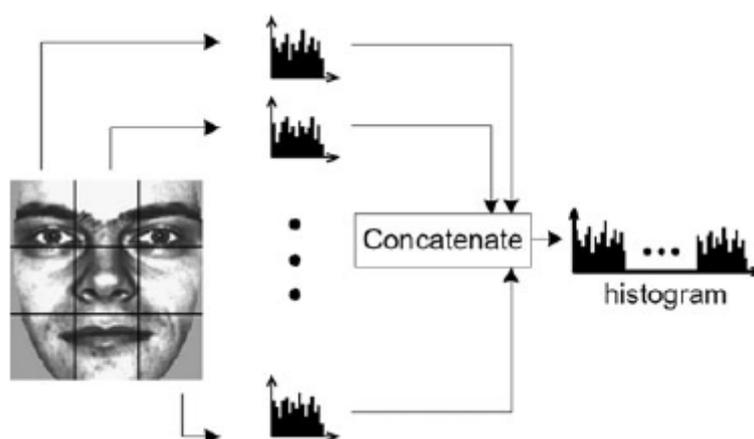


Figura 2.11: LBP para reconhecimento de face (Fonte: (PIETIKÄINEN et al., 2011, p. 152))

Esses métodos baseados em características locais parecem ser mais robustos às variações de posicionamento, e de iluminação, que os métodos referenciados como holísticos (PIETIKÄINEN et al., 2011, p. 152).

LBP para detecção de olhos O *LBP* tem sido utilizado para uma série de trabalhos objetivando detectar olhos nas imagens. Em vários trabalhos o *LBP* foi mesclado com outras abordagens, geralmente, os dados resultantes do descritor *LBP* servem de entrada para outros processamentos matemáticos, que tentam aumentar a acurácia da detecção desejada (PIETIKÄINEN et al., 2011, p. 153).

LBP para segmentar faces Devido à sua simplicidade computacional e ao seu poder discriminativo, o descritor *LBP*, recentemente, tem sido adotado para detectar faces humanas, em imagens de uma ou mais pessoas, mescladas com vários objetos, demonstrando excelentes resultados, que superam os resultados de vários trabalhos existentes, desenvolvidos com a



Figura 2.12: LBP para detecção de olhos (Fonte: (PIETIKÄINEN et al., 2011, p. 154))

mesma finalidade (PIETIKÄINEN et al., 2011, p. 154).

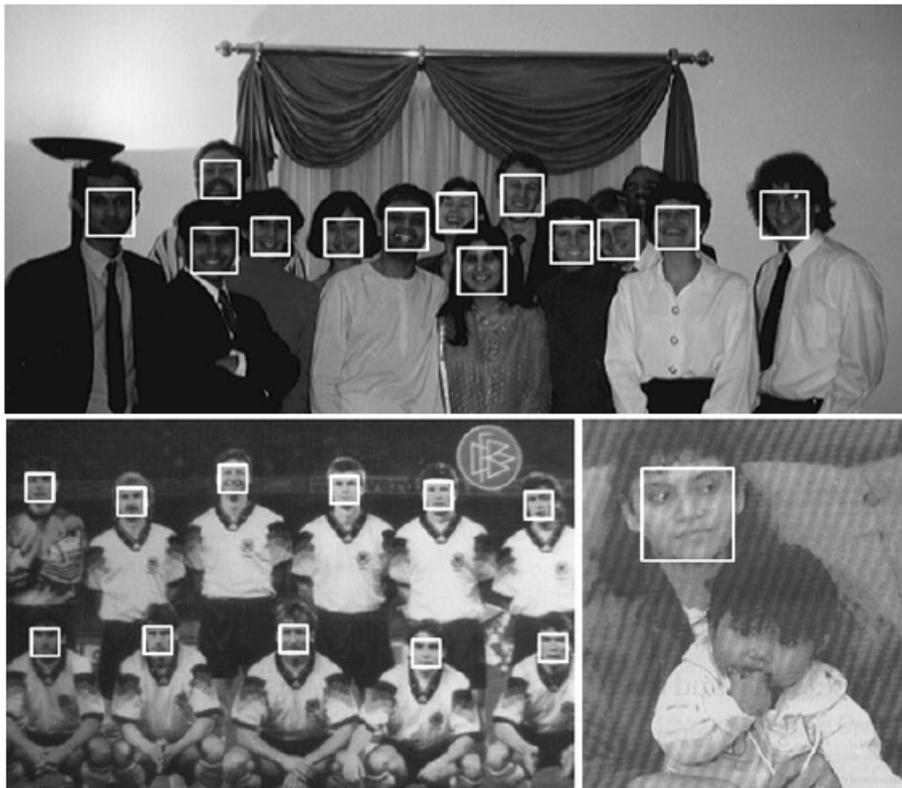


Figura 2.13: LBP para detecção de face (Fonte: (PIETIKÄINEN et al., 2011, p. 157))

Segundo os mesmos autores (Pietikäinen et al. (2011)), mesmo à despeito dos excelentes resultados percebidos quando o *LBP* é utilizado para detectar faces, ainda é possível encontrar problemas, como a face do bebê, exibida na Figura 2.13, que não foi detectada.

LBP para discernir emoção O operador *LBP* também tem sido utilizado para classificar a emoção expressada por indivíduos através de suas imagens: Os autores Pietikäinen et al. (2011, p. 164) relatam um experimento, que utilizava uma base de dados referenciada como *JAFFE* (*Japanese Female Facial Expression*), que possuía 213 imagens, nas quais, 10 pessoas diferentes expressam sentimentos diversos. Este teste, que fez uso do *LBP*, conseguiu atingir 93,8% de acerto. A Figura 2.14 mostra algumas imagens da base de dados *JAFFE*.



Figura 2.14: LBP para discernir emoção (Fonte: (PIETIKÄINEN et al., 2011, p. 164))

Segundo Pietikäinen et al. (2011, p. 165), a abordagem *LBP* também é adotada para uma série de outras finalidades, entre as quais :

- Reconhecimento de gênero, ou seja, informar se determinada imagem pertence a um homem ou a uma mulher;
- Mensuração de idade;
- Classificação de etnicidade;
- Identificação biométrica através da imagem da íris;
- Identificação de sorrisos em imagens.

2.1.7 Normalização

É possível necessitar trabalhar com dados, ou com conjuntos de dados que, pela sua própria natureza, possuam escalas de valores muito diferentes. Para exemplificar, imagine uma planilha com vários dados acerca dos pacientes de um hospital, entre esses dados, considere que a planilha possua duas colunas alusivas à altura e ao peso de cada paciente. A altura, naturalmente, deverá ter valores que oscilem entre 0,25 (tamanho de um bebê pré-maturo) e 2 (tamanho de um adulto considerado alto); já o peso, quase sempre, deverá oscilar entre 2,3 (peso de um bebê pré-maturo) e 250 (peso de um adulto com obesidade mórbida). Se esses dados fossem analisados sem as respectivas unidades de mensuração (altura e peso), isso poderia gerar a falsa impressão de que o valor 1,9 (referente à altura) estava muito baixo, se comparado ao valor 45 (referente ao peso). E, considerando as unidades de mensuração, a realidade é inversa: 1,9 é um valor alto e 45 é um valor baixo. Diante desse cenário, a pergunta chave é: como comparar valores alusivos a unidades de mensuração distintas?

Segundo Abdi (2010, p. 1), objetivando conseguir fazer tais comparações de forma justa, é necessário desconsiderar essas unidades de mensuração, através de um processamento matemático conhecido como normalização.

Ainda segundo Abdi (2010, p. 1), existem dois tipos principais de normalização:

- *O primeiro tem origem na álgebra linear*: Trata os dados como um vetor, em um espaço multidimensional, e, nesse contexto, normalizar os dados significa transformar o vetor de dados em um novo vetor cuja norma é um;
- *O segundo tem origem na estatística*: E consegue desconsiderar as unidades de mensuração por realizar um processamento matemático que transforma os dados considerados em um novo conjunto de dados (conhecido como *Z-scores*) que possui, obrigatoriamente, as seguintes características:
 - Média aritmética igual a 0;
 - Desvio padrão igual a 1.

É possível constatar a importância da normalização pesquisando trabalhos acadêmicos, como se verifica na seguinte citação:

“Durante o treinamento das redes MLP, verificamos a grande importância da normalização dos dados na qualidade dos resultados. A alteração da formatação

dos padrões de saída da rede da faixa entre 0 e 1 para a faixa 0.1 e 0.9, permitiu criar sistemas mais estáveis e com erro médio bem inferiores.”

(Fonte: (FERREIRA, 2004, p. 99))

Normalização para a “norma de um”

Na álgebra linear a norma de um vetor mensura seu tamanho, que é igual à distância euclidiana entre o seu ponto final e o seu ponto de origem. Esse cálculo é feito com base no teorema de pitágoras, entendendo que o vetor considerado pode ter N dimensões, e que o seu tamanho será igual à raiz quadrada da soma dos quadrados dos seus elementos (ABDI, 2010, p. 2).

Para ilustrar, com dados reais, como a normalização “norma de um” funciona, considere o exemplo visto abaixo:

Dado o vetor: $y = (35, 36, 46, 68, 70)$, sua norma seria denotada por $\|y\|$, e seria igual a:

$$\sqrt{35^2 + 36^2 + 46^2 + 68^2 + 70^2} = \sqrt{14161} = 119$$

O vetor normalizado, referenciado por \tilde{y} , seria igual a:

$$\tilde{y} = \left(\frac{35}{119}, \frac{36}{119}, \frac{46}{119}, \frac{68}{119}, \frac{70}{119} \right) = (0.2941, 0.3025, 0.3866, 0.5714, 0.5882)$$

Esse é o vetor normalizado segundo a normalização para a norma de um.

Essa normalização não se chama “norma de um” por acaso: a norma de um vetor normalizado é sempre igual a 1, conforme se observa, aproveitando os dados resultantes do exemplo acima.

$$\begin{aligned} \|\tilde{y}\| &= \sqrt{0.2941^2 + 0.3025^2 + 0.3866^2 + 0.5714^2 + 0.5882^2} = \\ &= \sqrt{0.0864 + 0.0915 + 0.1494 + 0.3264 + 0.3459} = \sqrt{1} = 1 \end{aligned}$$

Normalização Z-scores

Para entender os motivos práticos que criam a necessidade da normalização, analise a seguinte metáfora observada em (FREUND; SIMON, 2000, p. 64-65): Considere que o exame final, de

um curso de língua francesa, seja formado por uma prova de vocabulário e outra de gramática. Considere que um estudante obteve 66 pontos no exame de vocabulário e 80 pontos no exame de gramática. À primeira vista, parece que o resultado obtido no exame de gramática foi muito melhor, entretanto, considere que as notas dos alunos, que realizaram esses exames, possuem:

- *Para o exame de vocabulário:* Média = 51; Desvio Padrão = 12;
- *Para o exame de gramática:* Média = 72; Desvio Padrão = 16.

Com base nessas informações, é possível argumentar que a nota do estudante, referente ao exame de vocabulário, está a $\frac{66-51}{12} = 1,25$ desvios-padrão acima da média da turma. Também é possível afirmar que a nota do mesmo estudante, referente ao exame de gramática, está a apenas $\frac{80-72}{16} = 0,5$ desvios-padrão acima da média da turma. Nesse contexto, começa a ficar claro que a comparação direta das notas não é justa, pois, analisando o conjunto, está estatisticamente provado que a classificação do estudante, em relação à classificação da turma, foi melhor na prova de vocabulário e pior na prova de gramática.

O que acabou de ser feito foi uma conversão das notas em unidades padronizadas, também referenciadas como escores-z.

Abordando a questão formalmente, se x é uma mensuração pertencente a um conjunto de dados com média \bar{x} e desvio padrão s , então, seu valor em unidades padronizadas, denotadas por z , é (FREUND; SIMON, 2000, p. 65):

$$z = \frac{x - \bar{x}}{s} \quad (2.3)$$

2.1.8 *Instance Based Learning*

Segundo Mitchell (1997, p. 2), o aprendizado de máquina (*Machine Learning*) é, em função das suas próprias características, um campo multidisciplinar, que engloba:

- Inteligência artificial;
- Probabilidade e estatística;
- Teoria da complexidade computacional;
- Teoria do controle;
- Teoria da informação;
- Filosofia;
- Psicologia;
- Neurobiologia e outros campos.

Geralmente o aprendizado de máquina é categorizado como **aprendizado supervisionado** e **aprendizado não supervisionado**. No **aprendizado supervisionado** um “professor” provê um rótulo para cada elemento do conjunto de treinamento, e faz os ajustes necessários para diminuir o erro de classificação para cada elemento do conjunto. Já no **aprendizado não supervisionado** não existe um professor explícito, as amostras são tratadas sem a Figura de um rótulo associado, nesse caso, o sistema tenta “deduzir” os padrões existentes no conjunto observado (NISKANEN, 2003, p. 28).

Segundo Chapelle, Schölkopf e Zien (2006, p. 1-3) o que distingue as formas de aprendizado é a presença, ou ausência, do atributo *classe* que serve de rótulo para o conjunto de treinamento que será fornecido para o algoritmo: No **aprendizado supervisionado** cada instância do conjunto de treinamento terá seu próprio rótulo (*classe* associada); no **aprendizado não supervisionado** não existe a Figura do rótulo (*classe* associada); e, no **aprendizado semi-supervisionado** algumas instâncias do conjunto de treinamento terão rótulo (*classe* associada), e outras não.

O Quadro 2.1 representa uma forma muito comum de referenciar um conjunto de treinamento, com suas instâncias, atributos e *classes* associadas:

Quadro 2.1: Conjunto de treinamento com suas instâncias, atributos e classes

INSTÂNCIAS (I)	ATRIBUTOS (a)				CLASSES (C)
	Atributo 1	Atributo 2	...	Atributo M	
I_1	a_{11}	a_{12}	...	a_{1M}	C_1
I_2	a_{21}	a_{22}	...	a_{2M}	C_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
I_N	a_{N1}	a_{N2}	...	a_{NM}	C_N

Fonte: Adaptado de (FERRERO, 2009, p. 28)

Geralmente, o conjunto de treinamento, para um algoritmo supervisionado, consiste de um conjunto de instâncias I com N exemplos, isto é, $I = \{I_1, \dots, I_N\}$. Cada i -ésima instância I_i possuirá M atributos que a representará, assim, é possível dizer que I_i será representado por $\{a_{i1}, \dots, a_{iM}\}$. Adicionalmente, cada i -ésima instância I_i será rotulada com a sua classe correspondente C_i . Esse rótulo será fruto de uma função desconhecida f que determina o rótulo correspondente a cada instância ($C_i = f(I_i)$). Considerando que a função f é desconhecida, o papel do treinamento supervisionado será estabelecer uma lógica através da qual tal função (f) possa ser representada por uma função hipótese (*hip*), de tal forma que, pelo menos para a grande maioria dos casos, $C_i = f(I_i) = \text{hip}(I_i)$. Quando os rótulos possuem valores nominais (por exemplo, $C_1 = \text{'Quadrado'}$, \dots , $C_N = \text{'Círculo'}$) o processo realizado pela função *hip* é referenciado como classificação, e, quando os rótulos possuem valores numéricos (por exemplo, $C_1 = 10$, \dots , $C_N = 15$) o processo realizado pela função *hip* é referenciado como regressão (FERRERO, 2009, p. 28).

Ainda segundo Ferrero (2009, p. 28-29), os algoritmos supervisionados podem ser do tipo *eager* ou *lazy*. Os algoritmos do tipo *eager* utilizam os dados de treinamento para construir uma hipótese (*hip*), e, quando *hip* é construída, os exemplos de treinamento já podem ser descartados, pois, *hip* é suficiente para fazer as previsões. Já os algoritmos *lazy* não constroem uma hipótese: utilizam todos os exemplos de treinamento, que ficam guardados para essa finalidade, objetivando executar cada previsão requisitada.

Pelo que já foi exposto, é possível afirmar que os algoritmos utilizados para aprendizado baseado em instância (*Instance-Based Learning*) são algoritmos supervisionados do tipo *lazy*.

Contrastando com outros métodos de aprendizagem, que ajustam uma função de classificação através de várias iterações com os exemplos de treinamento, os algoritmos de aprendizagem baseados em instâncias apenas armazenam os exemplos de treinamento. A generalização

obtida, em função destes exemplos, é postergada até que uma nova instância precise ser classificada, ou seja, quando for necessário classificar um novo exemplo, todo o conjunto de treinamento armazenado será considerado na referida classificação (MITCHELL, 1997, p. 230).

Como exemplo de algoritmo de aprendizado baseado em instância é possível citar:

- *Nearest Neighbor*;
- *Locally Weighted Regression*;
- *Case-Based Reasoning*.

Segundo Mitchell (1997, p. 231), uma das desvantagens, do aprendizado baseado em instância, é que o custo da classificação de novas instâncias pode ser alto. Outra desvantagem, percebida principalmente na abordagem *Nearest Neighbor*, é que ela geralmente considera todos os atributos das instâncias, quando tenta encontrar exemplos similares na memória. Desta forma, se o que realmente importa, para a classificação almejada, são apenas algumas características específicas das instâncias, e se essas características (atributos) forem analisadas paralelamente com vários atributos que não são importantes para a classificação, então, as instâncias que são verdadeiramente mais similares podem ficar muito distantes no espaço n -dimensional considerado, e, por conseguinte, o algoritmo pode falhar.

O algoritmo de classificação utilizado durante toda a pesquisa foi o *Nearest Neighbor*, em função disso, dos algoritmos citados, como exemplos de algoritmo de aprendizagem baseada em instância, apenas ele é detalhado neste trabalho.

KNN (K-Nearest Neighbor)

Segundo Mitchell (1997, p. 231) o método baseado em instância mais básico é o *K-Nearest Neighbor*. Esse algoritmo assume que todas as instâncias correspondem a pontos em um espaço n -dimensional R^n . Os vizinhos mais próximos (*Nearest Neighbors*) de uma instância são definidos em termos da distância euclidiana padrão, que é definida por este autor conforme descrito na Equação 2.4 (MITCHELL, 1997, p. 232):

Considere uma instância arbitrária x descrita pelo vetor de características

$$x = \{a_1(x), a_2(x), \dots, a_n(x)\}$$

No qual $a_r(x)$ denota o valor do r – éximo atributo da instância x .

Então, a distância entre duas instância x_i e x_j é definida como $d(x_i, x_j)$, e,

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (2.4)$$

Além da distância euclidiana, na literatura que trata de *Machine Learning*, várias outras abordagens são consideradas para mensurar a distância entre dois pontos, em um espaço n -dimensional R^n , tais como as seguintes, observadas em (DUDA; HART; STORK, 2001, p. 188):

- *Métrica de Minkowski;*
- *Distância de Manhattan;*
- *Métrica de Tanimoto.*

No contexto do processamento digital de imagens, outras métricas são consideradas válidas. Uma métrica bastante utilizada, neste escopo, é a distância *kullback-Leibler Divergence*, que pode ser observada na Equação 2.5.

$$d_{KL}(Q, C) = \sum_{i=1}^n c_i \log \frac{c_i}{q_i} \quad (2.5)$$

Essa métrica (*Kullback-Leibler Divergence*) já foi tão explorada que dela sugeriram várias outras, como a *distância de Jeffrey* vista em (FELIPE, 2005, p. 37), e outras vistas em (PETROU; SEVILLA, 2006, p. 577) e (ROSENBERG; HEBERT; THRUN, 2007, p. 3).

Sobretudo, até o referido momento, a investigação explicitada tratou apenas dos algoritmos de mensuração de distância, que podem ser utilizados dentro do algoritmo *K-NN*. O algoritmo 2.1 detalha a descrição do algoritmo *K-NN* encontrada em (MITCHELL, 1997, p. 232).

Considere a função discreta de classificação $f : \mathbb{R} \rightarrow V$, com $V = \{v_1, \dots, v_s\}$ (representando os rótulos da classe)

• **Algoritmo de treinamento**

Para cada exemplo de treinamento $\{x, f(x)\}$, adicione o exemplo à lista de exemplos

• **Algoritmo de classificação**

Dada uma amostra x_q para ser classificada,

Entendendo que $x_1 \dots x_k$ denotam as k instâncias (dos exemplos) mais próximas de x_q

Retorne :

$$\hat{f}(x_q) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \delta(v, f(x_i))$$

No qual, $\delta(a, b) = 1$ se $a = b$ e $\delta(a, b) = 0$ para todos os outros casos.

Algoritmo 2.1: K-NN (Fonte: (MITCHELL, 1997, p. 232))

O valor de $\hat{f}(x_q)$, retornado por esse algoritmo como sua estimativa para $f(x_q)$, é apenas o valor mais comum de f , entre os K exemplos de treinamento mais próximos de x_q . Se o valor de K for igual a 1, então, esse algoritmo associará a $\hat{f}(x_q)$ (como sua estimativa para $f(x_q)$) o valor de $f(x_i)$, pelo fato de x_i ser a instância, do conjunto de treinamento, que está mais próxima de x_q . Quando K for maior que 1, o algoritmo associará a $\hat{f}(x_q)$ (como sua estimativa para $f(x_q)$) o valor que ocorrer com mais frequência entre os K vizinhos mais próximos no conjunto de treinamento (MITCHELL, 1997, p. 232).

Segundo Duda, Hart e Stork (2001, p. 183) a motivação principal para considerar a utilização do algoritmo *K-Nearest Neighbor* é a simples observação das probabilidades naturais: elementos com maior similaridade tendem a estar próximos.

Objetivando tornar a explanação acerca da Figura 2.15 mais didática, considere o Quadro 2.1, que explica que cada instância I_i será representada por M dimensões ($I_i = \{a_{i1}, \dots, a_{iM}\}$), considere também que o espaço referenciado anteriormente como n -dimensional R^n terá, neste exemplo específico, duas dimensões ($M = 2$), e que essas dimensões serão didaticamente referenciadas como *Atributo 1* e *Atributo 2*. A conclusão à qual o leitor chegará é simples: mudar o valor do K considerado pode ser suficiente para mudar o resultado do algoritmo, pois, conforme se verifica no exemplo específico da Figura 2.15, para $K = 1$, a i -ésima instância em análise (I_i)

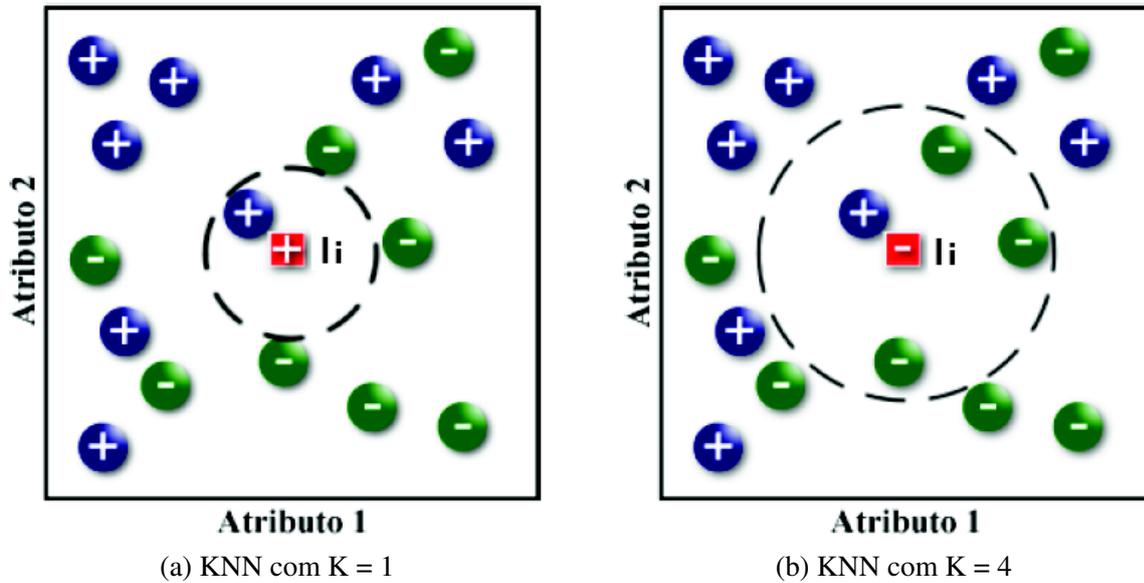


Figura 2.15: Variações de K no K -NN (Fonte: Adaptado de (FERRERO, 2009, p. 33))

foi classificada como positiva (pelo fato de seu vizinho mais próximo ser um elemento positivo), e, para $K = 4$, a i -ésima instância em análise (I_i) foi classificada como negativa (pelo fato deste ser o valor mais comum entre os 4 vizinhos mais próximos).

Segundo Duda, Hart e Stork (2001, p. 183) o algoritmo K -NN se comporta fazendo com que uma região esférica (imaginária) cresça até que os K elementos desejados sejam englobados, para que, à partir daí, o algoritmo possa “tirar suas conclusões”. A região esférica referenciada pode ser visualizada na Figura 2.15, na forma de tracejado circular.

Observe que a função de comparação é independente do algoritmo K -NN, ou seja, o algoritmo K -NN pode ser implementado com a distância Euclidiana, com a distância de *Manhattan*, com a distância *Kullback-Leibler Divergence* ou com outras. Além disso, a mudança do algoritmo de mensuração de distância pode alterar as taxas de acerto do K -NN, ainda que sejam considerados os mesmos conjuntos de treinamento e validação (leia a Seção 2.1.9), conforme pode ser verificado na Seção que trata das estatísticas alusivas aos testes realizados neste trabalho (seção 4.2). Para ilustrar melhor essa assertiva considere a Figura 2.16.

A ilustração da Figura 2.16 faz alusão a duas lojas hipotéticas (nas cores laranja e azul) localizadas numa cidade hipotética. Considere que os quadrados brancos são os quarteirões da cidade (medindo 50 metros x 50 metros), e que as linhas cinza representam as rodovias. A linha preta ilustra a *Distância Euclidiana*, e mede, aproximadamente, 212 metros. Todavia, se alguém quisesse se locomover entre essas duas lojas, na melhor das hipóteses, teria que

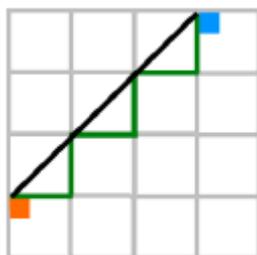
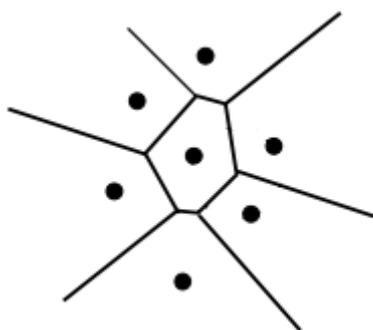


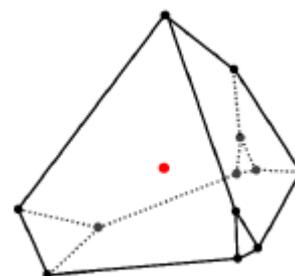
Figura 2.16: *Distância Euclidiana x Distância de Manhattan* (Fonte: (PUTTER, 2010, p. 36))

percorrer 300 metros, através do caminho tracejado com a cor verde, que seria calculado pela *Distância Manhattan* (PUTTER, 2010, p. 36).

Assim, para esse caso específico, a *Distância Manhattan* é a mais apropriada, entretanto, para cada aplicação é necessário analisar que métrica se encaixa melhor, já que as opções são variadas.



(a) Diagrama de *Voronoi* (Fonte: Adaptado de (AGATHOS, 2008, p. 5))



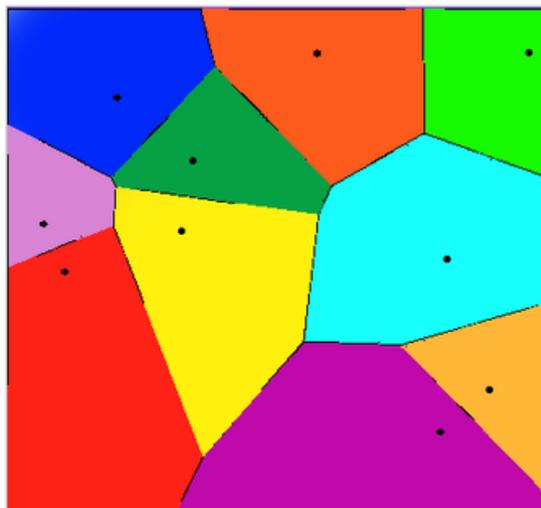
(b) Diagrama de *Voronoi* em 3D (Fonte: (PUTTER, 2010, p. 33))

Figura 2.17: Diagrama de Voronoi

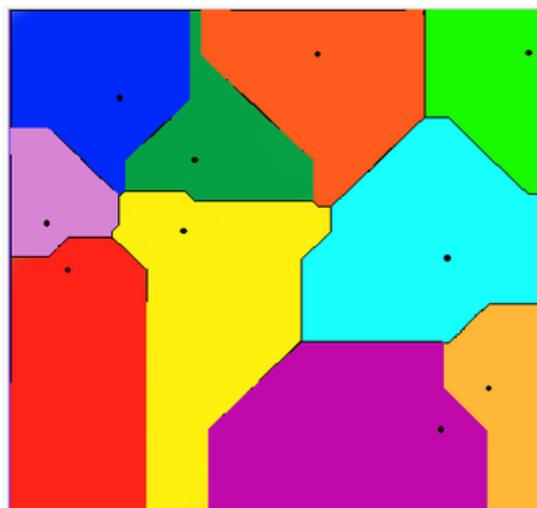
É possível visualizar, nas imagens acima, versões diferentes do diagrama conhecido como *Diagrama de Voronoi*. O diagrama visto na Figura 2.17a representa a superfície de decisão induzida pela execução de um algoritmo 1 – *NN*, utilizando a *Distância Euclidiana*, para um típico conjunto de exemplos de treinamento (AGATHOS, 2008, p. 5). O gráfico de *Voronoi* também pode ser visualizado em três dimensões (3D), conforme pode ser visto na Figura 2.17b.

Trabalhar com o gráfico em 2D costuma ser suficiente para fazer as inferências desejadas, além disso, trabalhar com o gráfico de *Voronoi* em 3D pode ser complicado em algumas situações específicas: Putter (2010, p. 34) criou uma seção, em sua tese, intitulada “*Why is the 3d voronoi diagram difficult to implement ?*”, desta forma, já é possível inferir que existem algu-

mas dificuldades associadas, inclusive, mesmo à despeito dessa assertiva, desenvolvendo um raciocínio lógico, sempre que uma dimensão é adicionada (em qualquer análise) adiciona-se o bônus de mais informações, e o ônus de mais elementos para entender/processar.



(a) Diagrama *Voronoi* calculado com *Distância Euclidiana*



(b) Diagrama *Voronoi* calculado com *Distância de Manhattan*

Figura 2.18: Variação do diagrama de *Voronoi* em função da métrica
(Fonte: http://en.wikipedia.org/wiki/Voronoi_diagram)

Por fim, o diagrama de *Voronoi* também depende, diretamente, da função adotada para mensuração de distâncias. Para explicar de uma forma mais didática, analise a Figura 2.18 com base na seguinte analogia: Imagine que existe uma cidade cujo território é um quadrado perfeito, então, os dois quadrados coloridos, vistos na referida Figura, são representações diferentes desta mesma cidade. Considere que cada ponto preto é uma loja, e que todas elas vendem, exatamente, os mesmos produtos, pelos mesmos preços e com a mesma qualidade de serviço. Imagine também que nenhuma das cidades vizinhas vendem esses mesmos produtos, de forma que, não compensaria viajar para uma cidade remota para comprar tais produtos. Assim, considere que o único fator analisado pelos clientes, na hora de comprar esses produtos, é a distância de onde eles se encontram, no momento que desejam comprar, até cada loja. Com base nesse preâmbulo, o espaço colorido, ao redor de cada loja, representa o espaço físico no qual se concentram seus clientes em potencial, posto que, dentro de cada Seção colorida, existe apenas uma loja, que é justamente a loja mais próxima, enquanto o cliente não atravessar a fronteira que o separa de uma região de outra cor.

Diante do que foi exposto, a Figura 2.18a orienta a escolha da loja, em função do posici-

onamento do cliente, com base na *Distância Euclidiana*. E a Figura 2.18b orienta a escolha da loja, em função do posicionamento do cliente, considerando o mesmo cenário, com base na *Distância de Manhattan*.

Assim, com base nessa metáfora e nos gráficos correlacionados, é possível perceber que a função de mensuração de distância influencia diretamente na formação do diagrama de *Voronoi*. Isso acontece porque os espaços de cada célula do gráfico são estabelecidos pela lógica da função de mensuração, conforme ratifica Putter (2010, p. 36).

Para encerrar, a Figura 2.19 mostra um exemplo do funcionamento do algoritmo *K-NN*.

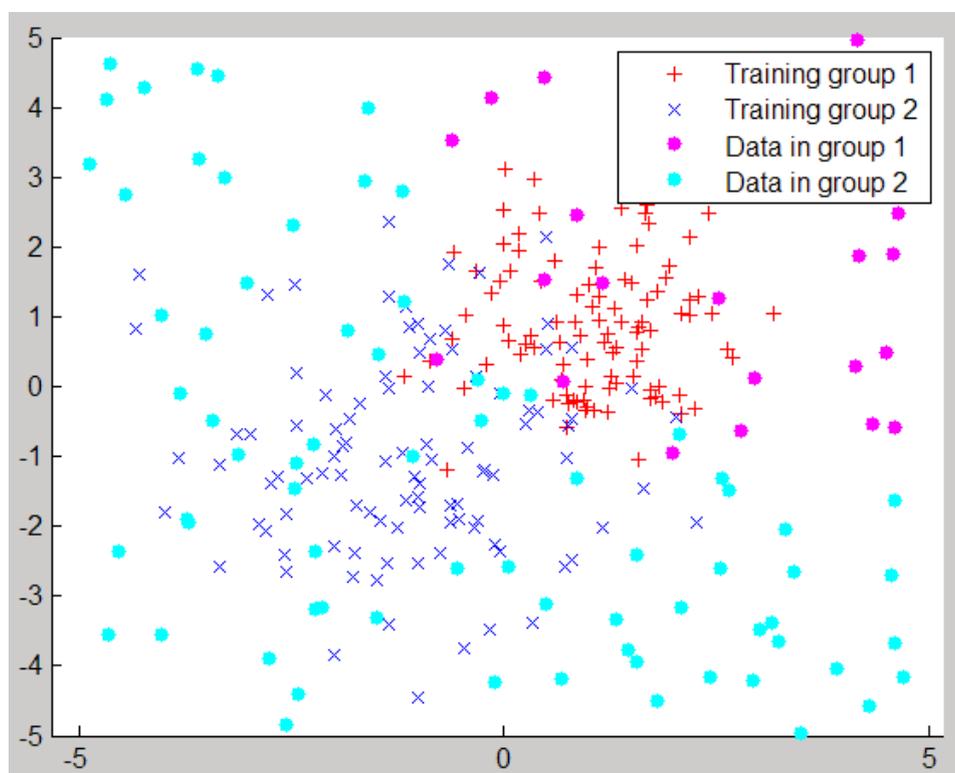


Figura 2.19: Resultado da execução do *K-NN* (Fonte: Adaptado do *help* do *Matlab*)

Nesta imagem (2.19), é possível enxergar todos os elementos do conjunto de treinamento (representados pelos símbolos + e x), e o agrupamento, do conjunto de validação, resultante da aplicação do algoritmo (representado pelos círculos • e •, conforme legenda inserida no canto superior direito da própria Figura). É possível perceber que a separação espacial entre os elementos classificados acontece em função do conjunto de treinamento, pois, no canto inferior esquerdo só temos círculos azuis (• - já que todos os vizinhos mais próximos são do tipo x), e no canto superior direito só temos círculos vermelhos (• - como consequência do fato de todos os vizinhos mais próximos serem do tipo +).

2.1.9 *Cross-Validation*

Cross-Validation, também referenciado como rotação de estimativas, é uma técnica estatística de particionamento de dados em subconjuntos desses mesmos dados, cujo objetivo é permitir que cada um desses subconjuntos possa ser utilizado para treinamento, e também para teste, em momentos distintos. As abordagens de *Cross-Validation* são frequentes em sistemas de aprendizagem de máquina (KOHAVI, 1995, p. 1138).

Segundo Duan, Keerthi e Poo (2002, p. 43) *Cross-validation* é uma técnica popular para estimar a generalização de erros, e existem várias versões da referida técnica. Na versão referenciada como *K-fold Cross-Validation* os dados de treinamento são aleatoriamente particionados em K subconjuntos mutuamente exclusivos (referenciados como *Folds* - daí o nome *K-fold Cross-Validation*) de tamanho aproximadamente igual.

O *Leave-One-Out Cross-Validation* é uma forma extrema de *Cross-Validation* na qual o K é igual ao número de exemplos do conjunto de treinamento, por isso se torna custoso computacionalmente e não é indicado para bases grandes de dados.

Ainda segundo Duan, Keerthi e Poo (2002, p. 43), tanto o *K-Fold Cross-Validation* quanto o *Leave-One-Out Cross-Validation* são utilizados com bastante frequência para mensurar o poder de generalidade de algoritmos de aprendizagem de máquina.

Segundo Tay (2008, p. 44) o *Leave-One-Out Cross-Validation* é particularmente importante para se testar bases de dados que ainda não possuem muitos dados, ou cuja obtenção de novos dados representa uma tarefa difícil.

Existem vários algoritmos de *Cross-Validation*. As subseções desta Seção vão citar os mais comuns. Sobretudo, neste trabalho, são utilizados apenas o *K-fold Cross-Validation* e o *Leave-One-Out Cross-Validation*, ambos resumidos neste breve preâmbulo.

Objetivos do Cross-Validation

Sumariamente, segundo Refaeilzadeh, Tang e Liu (2008, p. 1), o *Cross-Validation* é um método estatístico utilizado para avaliar e comparar algoritmos de aprendizagem. Também é possível afirmar que ele é uma técnica para estimativa de performance em modelos preditivos, que se baseia na postura de dividir os dados em dois segmentos:

- **Treinamento:** destinado ao processo de aprendizado/treinamento do modelo;

- **Validação:** destinado à etapa de testes no modelo já treinado.

Na prática, através das técnicas de *Cross-Validation*, os dados são alternadamente utilizados para treinamento e validação, ou seja, os dados que são utilizados para treinamento, em certa iteração, serão posteriormente utilizados para validação, e vice-versa. A forma mais comum de *Cross-Validation* é referenciada como *K-fold cross-validation*. As demais formas de *Cross-Validation* são casos especiais, ou envolvem repetidas iterações, do referido *K-Fold Cross-Validation* (REFAEILZADEH; TANG; LIU, 2008, p. 1).

Ainda segundo Refaeilzadeh, Tang e Liu (2008, p. 1), as técnicas de *Cross-Validation* possuem dois objetivos:

- Estimar a performance do modelo aprendido à partir dos dados disponíveis, ou seja, perceber a generabilidade alcançada por um algoritmo;
- Comparar a performance de dois ou mais algoritmos, e verificar qual deles funciona melhor para os dados disponíveis, ou, ainda, comparar a performance de duas ou mais variantes de um modelo parametrizado.

Em função dos objetivos acima, vários procedimentos já foram propostos. Entre os mais conhecidos, é possível citar:

- *Resubstitution Validation*;
- *Hold Out Validation*;
- *K-Fold Cross-Validation*;
- *Leave-One-Out Cross-Validation*;
- *Repeated K-Fold Cross-Validation*.

Resubstitution Validation

Segundo Refaeilzadeh, Tang e Liu (2008, p. 2), nessa abordagem o modelo utiliza todos os dados para treinamento, e, posteriormente, utiliza os mesmos dados (todos) para teste. Essa postura, diversas vezes, acaba resultando em *overfitting*: situação caracterizada pelo fato do

algoritmo responder à contento para os dados utilizados durante o treinamento, e, surpreendentemente, responder de forma falha para dados não vistos no conjunto de treinamento, ou seja, o sistema perde o poder de generalizar, e parece ter simplesmente ”decorado“ algumas amostras utilizadas na fase de treinamento.

Hold-Out Validation

Segundo Refaeilzadeh, Tang e Liu (2008, p. 2), objetivando evitar o *overfitting*, um conjunto independente de dados é selecionado para treinamento, e outro conjunto independente de dados é selecionado para validação, e eles não são permutados. Isso resulta em uma estimativa com maior acurácia para a performance de generalização do algoritmo. O lado negativo desta abordagem é que ela não utiliza todos os dados disponíveis para treinamento, nem para validação, e, por conseguinte, os resultados são diretamente dependentes da seleção feita entre dados de treinamento e dados de validação. Para tratar desses desafios, e maximizar a utilização dos dados disponíveis, a abordagem *K-Fold Cross-Validation* é indicada.

Adicionalmente, segundo Kohavi (1995, p. 1139) é comum na abordagem *Hold-Out Validation* utilizar $\frac{2}{3}$ dos dados como conjunto de treinamento, e $\frac{1}{3}$ dos dados como conjunto de teste. E, é válido enfatizar, ainda segundo o mesmo autor (KOHAVI, 1995, p. 1139), que alguns autores nem referenciam o *Hold-Out Validation* como um método de *Cross-Validation*, posto que os dados utilizados nesta abordagem não são cruzados em nenhum momento: os dados utilizados para validação serão utilizados apenas para validação, e os dados utilizados para treinamento serão utilizados apenas para treinamento.

K-Fold Cross-Validation

Ainda segundo Refaeilzadeh, Tang e Liu (2008, p. 2), nesse tipo de validação, o primeiro passo é particionar os dados em K segmentos (ou *Folds*) iguais (ou aproximadamente iguais). Em seguida, K iterações de treinamento e validação são realizadas, da seguinte forma: a cada iteração, um dos segmentos (*Folds*) é separado para validação dos dados, e os outros $K - 1$ segmentos (*Folds*) são utilizados para treinamento. A Figura 2.20 mostra um exemplo considerando $K = 3$:

Segundo Syed (2011, p. 11) o *K-Fold Cross-Validation* é uma alternativa muito interessante porque balanceia o custo computacional e o incremento na estimação de resultados.



Figura 2.20: *K-Fold Cross-Validation* com $K = 3$

Antes de particionar os dados em K segmentos (*Folds*) é muito comum estratificar os dados. A referida estratificação é o processo de rearranjar os dados, para garantir que cada segmento (*Fold*) tenha uma boa representatividade acerca de todas as classes que devem ser considerados durante o processo de aprendizagem. Por exemplo, em problemas de classificação binária (onde cada classe representa cerca de 50% dos dados) é conveniente arranjar cada segmento (*Fold*) de modo que ele possua, mais ou menos, metade das instâncias sendo de uma classe, e metade sendo da outra (REFAEILZADEH; TANG; LIU, 2008, p. 2).

Kohavi (1995, p. 1143) comparou várias abordagens para estimar acurácia (incluindo *Cross-Validation* sem estratificação, *Leave-One-Out Cross-Validation*, *Cross-Validation* com estratificação, além de outras técnicas) e recomendou a utilização de *10-Fold Cross-Validation* como a melhor abordagem para quem almeja mensurar acurácia ou trabalhar com seleção de modelo preditivo. Sobretudo, à despeito da afirmação do autor, esse sentimento dele (acerca do que é melhor) se aplica, especificamente, às sete bases de dados utilizadas nos testes que ele fez, e, especificamente, aos classificadores *C4.5* e *Naive-Bayes*, que ele utilizou. Assim, parece lógico que o *10-Fold Cross-Validation* estratificado não é uma verdade universal que possa ser aplicada a qualquer situação. Salzberg (1997, p. 325) também recomenda trabalhar com o $K = 10$, mas, enfatiza que essa "situação ideal" pode deixar de ser a "ideal" à depender da base. Adicionalmente, Diettrich (1998, p. 1922) informa que o *10-Fold Cross-Validation* não é a solução para tudo, inclusive, recomenda outras abordagens para algumas situações específicas.

Leave-One-Out Cross-Validation

O *Leave-One-Out Cross-Validation* (*LOOCV*) é um caso especial do *K-Fold cross-validation*, no qual o K é igual ao número de instâncias de dados, ou seja, a cada iteração, todos os dados disponíveis, com exceção de um deles, serão utilizados para o treinamento. Por conseguinte, a cada iteração, o modelo será testado apenas uma vez: com o k -ésimo elemento que não

esteve presente no conjunto de dados de treinamento. Essa técnica é indicada, principalmente, quando os dados disponíveis para treinamento/teste são raros, como nos estudos relativos à bioinformática, nos quais, os dados disponíveis não extrapolam algumas dúzias de amostras (REFAEILZADEH; TANG; LIU, 2008, p. 3).

Repeated K-Fold Cross-Validation

Para obter estimativas mais confiáveis de performance é sempre preferível aumentar a quantidade de testes. Fazendo uso do tradicional *K-Fold Cross-Validation* apenas K testes serão executados. Uma forma usada com frequência para aumentar essa quantidade de testes é executar o *K-Fold Cross-Validation* repetidas vezes, considerando uma reorganização, e re-estratificação, dos dados, antes de cada iteração. Essa abordagem é referenciada como *Repeated K-Fold Cross-Validation* (REFAEILZADEH; TANG; LIU, 2008, p. 3).

2.1.10 Ruído em imagens digitais

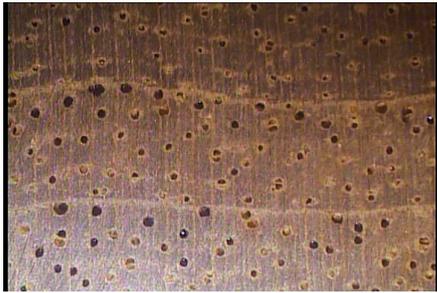
Qualquer propriedade, de um objeto em análise, que seja fruto de fatores externos aleatórios, e que possa atrapalhar a sua interpretação ou classificação, pode ser referenciado como ruído. Todo problema não trivial, na área de *Machine Learning*, notadamente no escopo do reconhecimento de padrões, envolve algum nível de ruído (DUDA; HART; STORK, 2001, p. 12).

Segundo Gonzalez e Woods (2010, p. 205), as principais fontes de ruído, em imagens digitais, surgem durante o processo de aquisição (abordado na Seção 2.1.4) ou durante o processo de transmissão (abordado na Seção 2.1.4). Vários fatores podem atuar como fontes de ruído, entre eles, é possível citar:

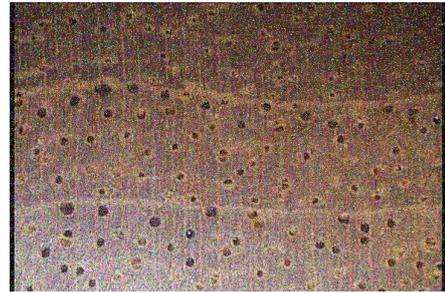
- A qualidade dos sensores de aquisição de imagem;
- As condições ambientais no momento da aquisição;
- Interferências no canal utilizado para transmissão.

Para perceber o efeito de um ruído, em uma imagem digital, observe a Figura 2.21.

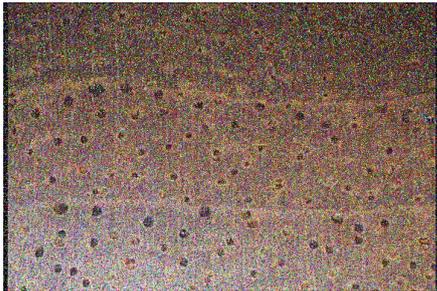
Para exemplificar, sem utilizar imagens de madeira, observe a Figura 2.22, que exibe o resultado da execução de um código que adiciona 2% de ruído a determinada imagem.



(a) Imagem de Cedro original (sem ruído)



(b) Imagem de Cedro com 15% de ruído



(c) Imagem de Cedro com 25% de ruído



(d) Imagem de Cedro com 55% de ruído

Figura 2.21: Imagens de Cedro com ruído gerado pelo *Matlab* (Fonte: Adaptação de imagens fornecidas pela *Embrapa*)



(a) Imagem original (sem ruído)



(b) Imagem original com 2% de ruído

Figura 2.22: Resultado de inserção de ruído no *Matlab* (Fonte: Adaptado do *help* do *Matlab* - <http://www.mathworks.com/help/toolbox/images/ref/imnoise.html>)

2.2 Soluções alternativas

Existem outras alternativas para atenuar a tráfico ilegal de madeiras no Brasil, seria possível, por exemplo:

- **Fiscalizar, de forma mais eficiente, a derrubada das árvores que estão protegidas por lei:** se as árvores não fossem derrubadas os agentes alfandegários não precisariam se preocupar com a passagem delas pelas alfândegas;
- **Capacitar e remunerar melhor os agentes alfandegários:** A capacitação seria direcionada para a identificação mais eficiente das madeiras que são alvos mais frequentes do tráfico ilegal; E uma remuneração melhor iria atenuar a corrupção que existe em tais alfândegas;
- **Investir em políticas de conscientização:** Objetivando criar prosélitos para a causa, que seriam capazes de denunciar os infratores subsidiando um combate mais efetivo;
- **Reformular as leis do país:** impondo penas mais rígidas para os atores deste comércio ilegal;
- **Proibir a comercialização, dentro do Brasil, de produtos cuja matéria prima seja qualquer madeira proibida por lei:** Criando esquemas de fiscalização que atuassem nas empresas que entregam os produtos finais aos consumidores, identificando produtos construídos com matéria prima ilícita;
- **Contratar um dos produtos identificados na Seção Trabalhos relacionados (seção 2.3):** com o intuito de atender imediatamente as necessidades, solicitando, se necessário, as devidas alterações para a realidade do Brasil.

Parece sensato afirmar que nenhuma dessas "soluções", aplicada de forma isolada, resolveria definitivamente o problema. Tudo indica que se trata de um problema eminentemente cultural, e que o investimento em educação é primordial para evitar que as gerações futuras continuem cometendo o mesmo erro. Sobretudo, as alternativas acima citadas, utilizadas preferencialmente de forma conjunta, podem atenuar de maneira significativa a situação do tráfico ilegal de madeiras no Brasil.

2.3 Trabalhos relacionados

Como resultado da pesquisa por trabalhos relacionados, foi encontrado o trabalho de (KHALID et al., 2008) cujo título é: "Design of an intelligent wood species recognition system". Esse trabalho utiliza redes neurais para classificar amostras de madeiras, e a metodologia utilizada está resumida na Figura 2.23.

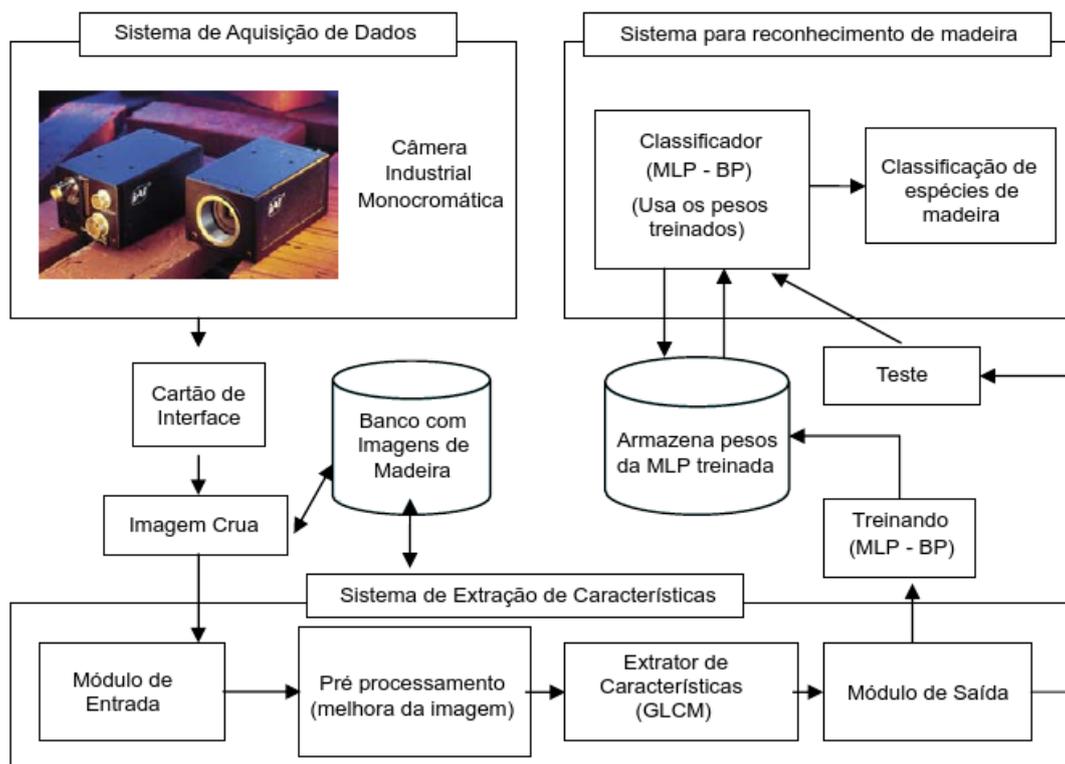


Figura 2.23: Sistema para reconhecimento de madeira (Fonte: (KHALID et al., 2008, p. 11))

Os dados são capturados através de uma câmera monocromática industrial, e são armazenadas em um banco de dados de imagens. Existe um sistema específico para extração das características que servirão de entrada para o treinamento da rede neural. A rede neural utilizada neste trabalho é a *MLP (MultiLayer Perceptron)*, e o algoritmo utilizado para o treinamento da rede é o *backpropagation*. Segundo o autor do referido trabalho, tanto o *MLP* como o *backpropagation* são bem detalhados em muitas literaturas acerca de *Machine Learning* (KHALID et al., 2008, p. 13).

Também foi encontrado o trabalho de (BREMANANTH; NITHYA; SAIPRIYA, 2009), intitulado "Wood Species Recognition System". Esse trabalho utiliza ferramentas científicas similares às dos autores Khalid et al. (2008): Faz uso de redes neurais do tipo *MLP* com o

algoritmo *backpropagation*.

Sem muito esforço, é possível encontrar vários trabalhos relacionados, tais como:

- **Autoria:** (TOU; LAU; TAY, 2008), **Trabalho:** *Computer Vision-based Wood Recognition System*;
- **Autoria:** (TOU; TAY; LAU, 2009a), **Trabalho:** *A Comparative Study for Texture Classification Techniques on Wood Species Recognition Problem*;
- **Autoria:** (QIU, 1999), **Trabalho:** *A Simple Machine Vision System for Improving the Edging and Trimming Operations Performed in Hardwood Sawmills*;
- **Autoria:** (HEDAYATI; CHONG; KEUTZER, 2010), **Trabalho:** *Recognition of Tibetan Wood Block Prints with Generalized Hidden Markov and Kernelized Modified Quadratic Distance Function*;
- **Autoria:** (KAUPPINEN, 1999), **Trabalho:** *Development of a color machine vision method for wood surface inspection*;
- **Autoria:** (WANG, 2010), **Trabalho:** *Wood recognition based on grey-level co-occurrence matrix*;
- **Autoria:** (NASIRZADEH, 2010), **Trabalho:** *Woods Recognition System Based on Local Binary Pattern*;
- **Autoria:** (TOU; TAY; LAU, 2009b), **Trabalho:** *Rotational Invariant Wood Species Recognition through Wood Species Verification*;
- **Autoria:** (PAN; KUDO, 2012), **Trabalho:** *Recognition of Wood Porosity Based on Direction Insensitive Feature Sets*.

A maioria dos trabalhos relacionados tem origem, e aplicabilidade prática, na Malásia, pois, o referido país está entre os onze maiores exportadores de madeira serrada do mundo, e, os índices de tráfico ilegal de madeira, no referido país, são muito altos. Diante desse cenário, existe uma motivação natural para pesquisas relacionadas ao reconhecimento automatizado da madeira (PETRAUSKI et al., 2011, p. 103).

Nesses trabalhos relacionados, foi possível perceber que uma série de recursos distintos foram utilizados para análise/classificação de madeira:

- No trabalho de (PAN; KUDO, 2012) são utilizadas árvores de decisão (adotando o algoritmo C4.5);
- O trabalho de (MAENPAA, 2003) adota o LBP como ferramenta;
- O trabalho de (BREMANANTH; NITHYA; SAIPRIYA, 2009) utiliza o *GLCM (Grey-level Co-occurrence Matrices)*;
- O *paper* publicado por (TOU; LAU; TAY, 2008) mostra o estudo de resultados experimentais, que são frutos da combinação do *GLCM (Grey-level Co-occurrence Matrices)* com redes *MLP (Multi-layer Perceptron)*, e resultaram numa acurácia de 72%;
- Os mesmos autores (TOU; TAY; LAU, 2009a) publicaram, no ano seguinte, um *paper* comparando técnicas de classificação diferentes:
 - Matrizes *GLCM (Grey-level Co-occurrence Matrices)*, atingindo a acurácia de 78.33%;
 - Filtros *Gabor*, atingindo a acurácia de 73.33%;
 - Matrizes *GLCM (Grey-level Co-occurrence Matrices)* combinadas com filtros *Gabor*, atingindo a acurácia de 76.67%;
 - Matriz de covariância produzida através das características geradas pelos filtros *Gabor*, atingindo a acurácia de 85%.

Embora vários trabalhos relacionados tenham sido identificados, não foi encontrado, na pesquisa realizada, nenhum trabalho que combinasse *LBP* com *K-NN* (abordagem do método proposto neste trabalho) para atingir objetivos similares ao desta pesquisa (classificação de madeira).

CAPÍTULO 3

MÉTODO PROPOSTO

Esse capítulo descreve o método proposto, para classificação de madeiras, em três etapas que se relacionam, como pode ser visto na Figura 3.1. Nessa Figura é possível perceber que existem processos comuns entre as etapas: Criação da imagem digital; e Extração de características.

MÉTODO PROPOSTO

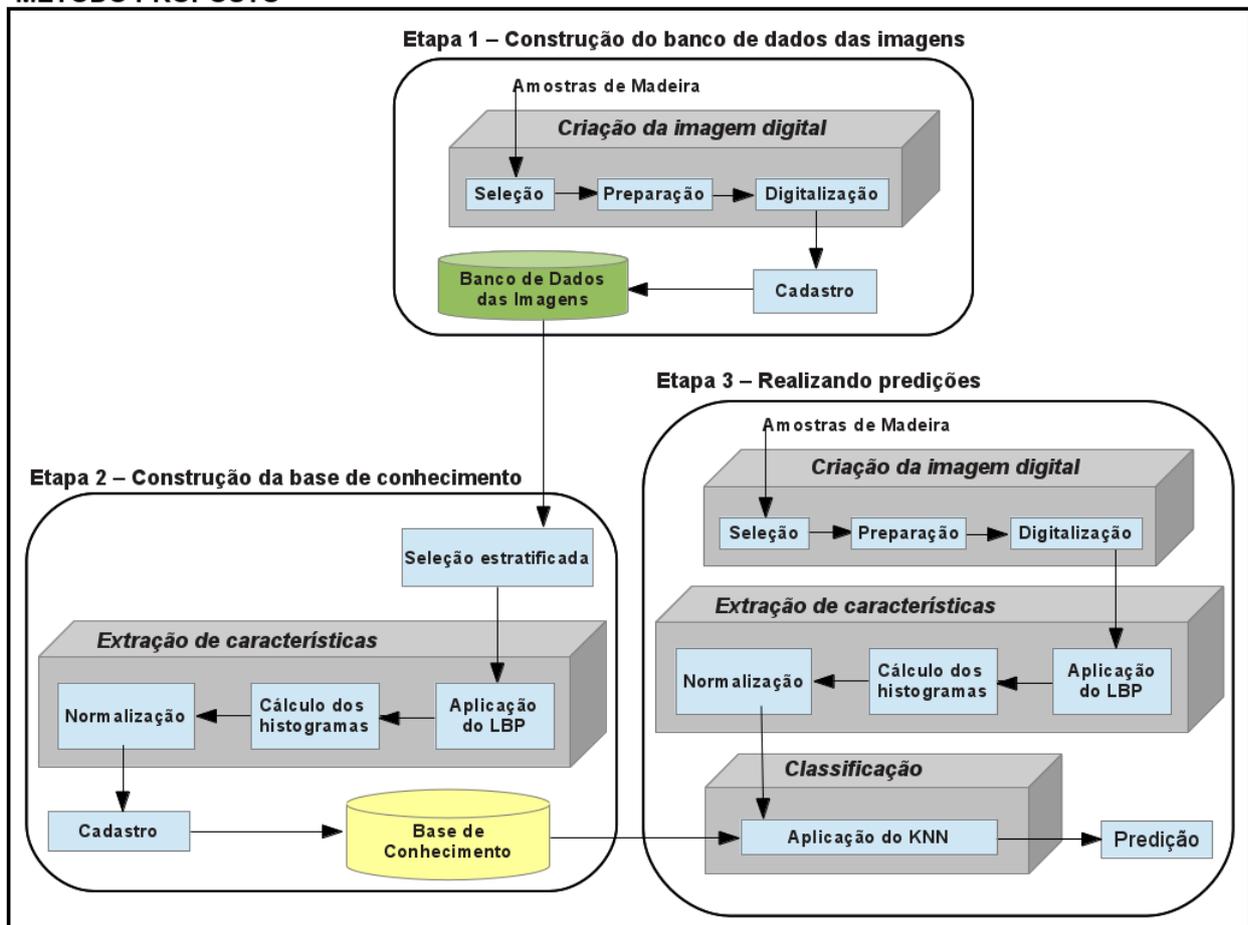


Figura 3.1: Fluxo geral do método de classificação de madeira proposto

Os processos comuns (Criação da imagem digital e Extração de características) serão abordados primeiro, na Seção 3.1. No processo de Criação da imagem digital serão explanados

todos os passos necessários para a adequada digitalização das amostras de madeira, considerando as necessidades do modelo proposto. No processo de Extração de características serão discutidas as tarefas que precisam ser executadas para calcular, e normalizar, os histogramas que representarão cada imagem. A Etapa 1, que detalha como o Banco de Dados das Imagens é construído, será explicada na Seção 3.2. A Etapa 2, que tem como objetivo a construção da Base de Conhecimento, será tratada na Seção 3.3. E, por fim, a Etapa 3, responsável pelas previsões do modelo proposto, será explanada na Seção 3.4.

3.1 Processos comuns entre as etapas

Os processos comuns serão explicados, antes de entrar no mérito de cada etapa do método proposto, com o objetivo de evitar a repetição das explicações. O processo de *criação da imagem digital* é executado na *Etapa 1* e na *Etapa 3*. E o processo de *Extração de características* é executado na *Etapa 2* e na *Etapa 3*, conforme exposto na Figura 3.1.

3.1.1 Criação da imagem digital

Esse processo se divide em três processos internos:

- Seleção;
- Preparação;
- Digitalização.

Seleção

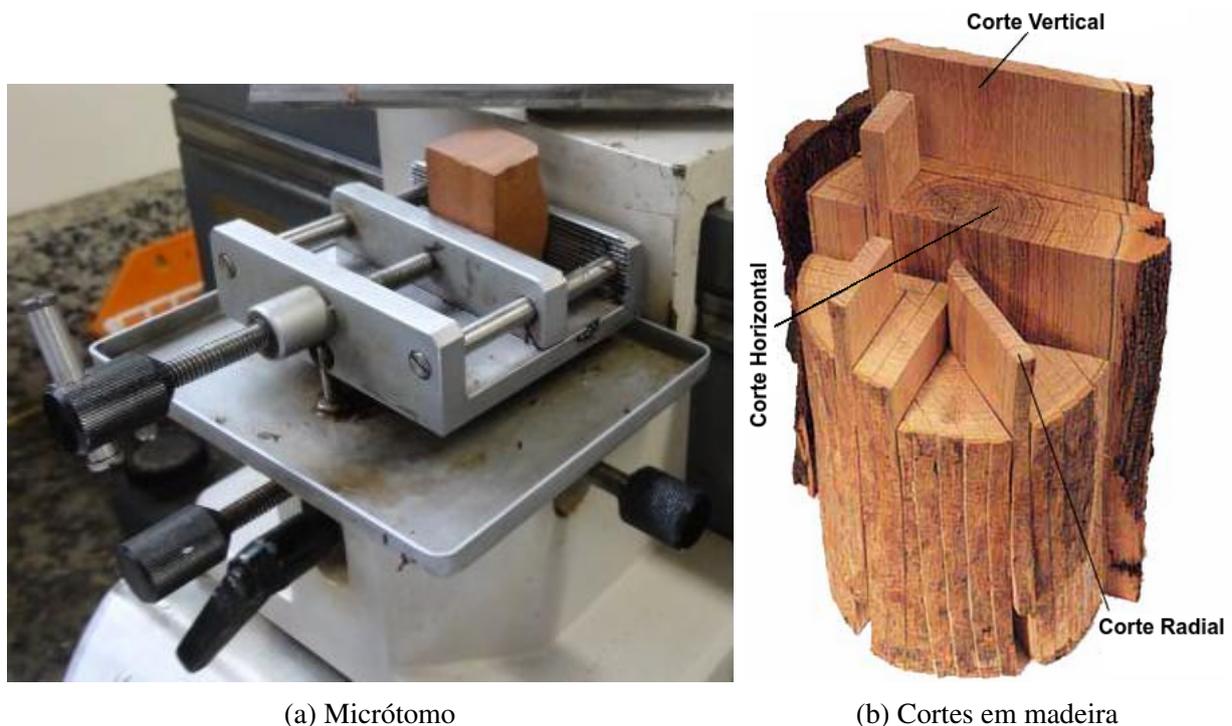
Conforme mencionado, o referido processo é realizado em duas etapas distintas (*Etapa 1* e *Etapa 3*). Essas seleções possuem objetivos diferentes: a seleção realizada na *Etapa 1* (*construção do banco de dados das imagens*) é feita com base nas classes da madeira que devem estar disponíveis no Banco de Dados das Imagens, que será utilizado como insumo para construção da Base de Conhecimento. A seleção realizada na *Etapa 3* (*realizando previsões*) é feita considerando a amostra de madeira que precisa ser submetida para o método, que retornará uma previsão acerca dela. E essa previsão falhará, se a seleção realizada na *Etapa 3* não

apresentar, como resultado, classes de madeira que existam no universo de classes da Base de Conhecimento.

Desta forma, as amostras resultantes da seleção realizada na *Etapa 1* podem ser distintas das amostras resultantes da seleção realizada na *Etapa 3*, entretanto, essas seleções devem elencar, como universo de possibilidades, as mesmas classes de madeira, sob pena de erros durante a predição. Assim, o método não conseguirá fazer predições acerca de classes para as quais não foi treinado. Daí a necessidade do sincronismo, agora descrito, entre as seleções que acontecem em etapas distintas.

Preparação

Este processo também é realizado na *Etapa 1* e na *Etapa 3*. Em ambas as etapas o objetivo é o mesmo: trata-se de um processo físico que deve ser executado sob a superfície da madeira selecionada, antes da sua digitalização. Esse processo corresponde a um corte, feito com um aparelho cortante, como uma faca, na superfície a ser digitalizada. Ele também pode ser feito com um aparelho concebido para este fim, como o micrótomo, visto na Figura 3.2a.



(a) Micrótomo

(b) Cortes em madeira

Figura 3.2: Micrótomo; Cortes em madeira (Fonte: *Embrapa*; Adaptação de <http://www.pauleiraguitars.com/materias/tipos-de-cortes-de-madeira>)

O processo de preparação da madeira deve ser exatamente o mesmo nas duas etapas nas quais se repete. Observe na Figura 3.2b que, a depender do tipo de corte estabelecido, a madeira exibe imagens diferentes.

Se as amostras preparadas na *Etapa 1* e na *Etapa 3* não possuírem os mesmos tipos de corte, o método retornará predições equivocadas na *Etapa 3*, pois essa diferença, referente ao tipo de corte, pode fazer com que a imagem, de uma mesma classe de madeira, se apresente de forma bem diferente.

Adicionalmente, o tipo de objeto utilizado para cortar a madeira também deve ser o mesmo em todas as etapas, pois um micrótomo, um machado e uma faca, aparentemente, fariam cortes diferentes na madeira, posto que cada um deles possui um tipo de lâmina diferente. Caso os cortes sejam diferentes, as imagens resultantes podem não ser similares para o modelo proposto.

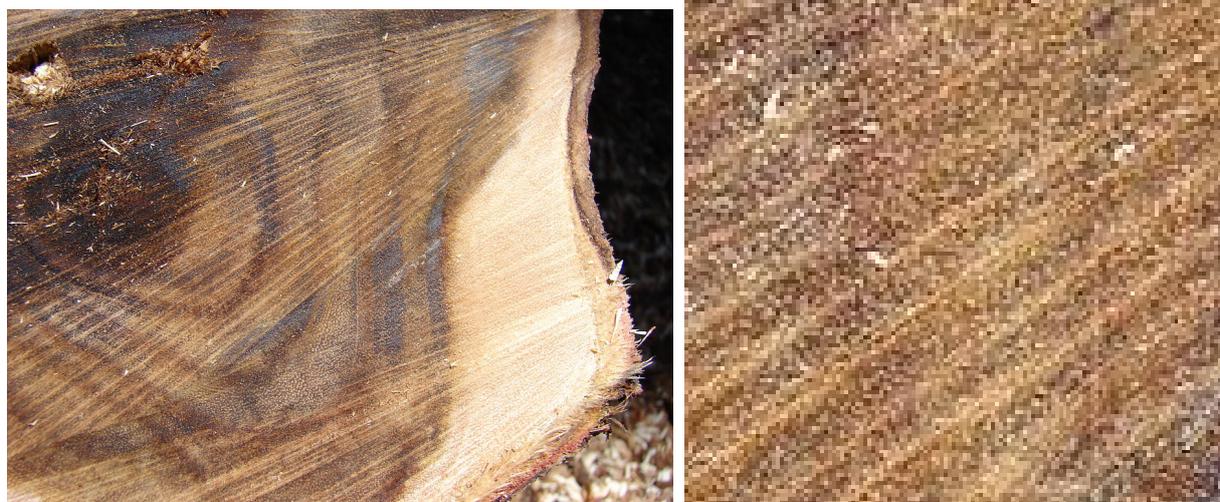
Por fim, um processo de limpeza se faz necessário, antes do processo de digitalização, pois, quando a madeira é cortada, algumas vezes, aparecem resquícios de farpas, como pode ser visto na Figura 3.3a.

Como a predição retornada pelo método depende diretamente das imagens geradas após o processo de preparação, é imprescindível garantir a sua uniformidade. Ou seja, sempre que esse processo se repetir é necessário que:

- O mesmo tipo de objeto cortante seja utilizado;
- O mesmo tipo de corte seja feito;
- Uma limpeza seja realizada na superfície que será digitalizada.

Digitalização

No escopo do modelo proposto, consiste na captura de uma imagem, de uma amostra de madeira, após os processos de seleção e preparação. O referido processo também acontece na *Etapa 1* e na *Etapa 3*, e também exige uniformidade. Observe as Figuras 3.3a e 3.3b.



(a) Madeira Acácia

(b) Zoom em madeira Acácia

Figura 3.3: Corte em acácia; Zoom no corte da acácia (Fonte: Adaptação de http://pt.wikipedia.org/wiki/Ficheiro:Acacia_koaia_wood.jpg)

As Figuras 3.3a e 3.3b são frutos de uma mesma amostra. Conforme explanado no referencial teórico, na Seção que trata de textura (Seção 2.1.2), a textura que se observa em uma imagem também é fruto da distância à partir da qual ela foi capturada. A Figura 3.3b é apenas o resultado de uma aproximação na Figura 3.3a.

Desta forma, caso as imagens digitalizadas na *Etapa 1* e na *Etapa 3* não sejam capturadas considerando distâncias iguais, ou pelo menos similares, o método proposto pode não atingir os objetivos, pois, a distância considerada pode mudar o aspecto da textura em análise.

Outro elemento que deve ser verificado, durante o processo de digitalização, é a luminosidade. Considere que existem seis objetos idênticos, ao redor da única fonte de iluminação de determinado ambiente, conforme retrata a Figura 3.4.

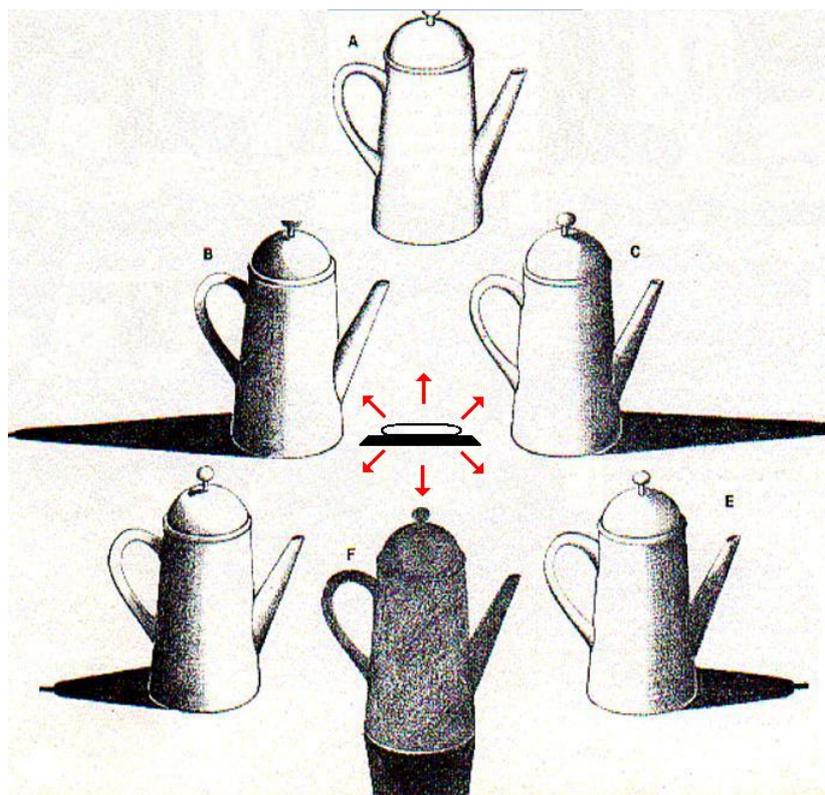


Figura 3.4: Efeito da luminosidade (Fonte: Adaptado de http://artesatividades.blogspot.com.br/2011/04/luzsombrapropriaesombraprojetada_09.html)

Entendendo que o objetivo é digitalizar um deles, como os seis objetos são idênticos, em princípio, qualquer um deles poderia ser digitalizado, entretanto, em função da luminosidade disponível, o objeto A é o objeto cuja digitalização trará melhores resultados. Considerando, especificamente, as digitalizações que devem ser feitas no método proposto, a sugestão é utilizar a luz solar, e deixar a superfície que deve ser digitalizada, após os processos de seleção e preparação, voltada para a luz solar.

3.1.2 Extração de características

Este processo se repete na *Etapa 2* e na *Etapa 3*. A base de conhecimento desse método, enxergada na *Etapa 2* do fluxo geral (disposto na Figura 3.1), é formada por uma tabela de dados que possui apenas duas colunas: uma com as características que são capturada de cada amostra, e outra com as respectivas classes. Adicionalmente, antes de qualquer predição, realizada na *Etapa 3*, é necessário extrair as características da imagem cuja classe será identificada.

O processo de extração das características é formado por três processos internos:

- Aplicação do *LBP*;
- Cálculo dos histogramas;
- Normalização.

Aplicação do *LBP*

O *LBP* descrito no referencial teórico (Seção 2.1.6) é o *LBP* básico. E o que foi utilizado nesse trabalho representa uma derivação do *LBP* básico, desenvolvida pelos autores Pietikäinen et al. (2011)

Para entender o *LBP* utilizado neste trabalho, conforme explanado por (PIETIKÄINEN et al., 2011, p. 14-16), considere que I é uma imagem monocromática (em tons de cinza) e g_c denota o nível de cinza de um *pixel* arbitrário (x,y) , ou seja, $g_c = I(x,y)$.

Adicionalmente, considere que g_p denota o valor de cinza de uma amostra de ponto em uma vizinhança circular. Considere que, nesta vizinhança, os espaçamentos entre os pontos são uniformes, isto é, o espaço entre um ponto e seus vizinhos é sempre igual. A referida vizinhança é formada por P amostras de ponto, e raio R ao redor do ponto central (x,y) .

A Figura 3.5 mostra três exemplos, considerando variações no tamanho da vizinhança (P) e no raio considerado (R), que ajudam a entender o esquema descrito.

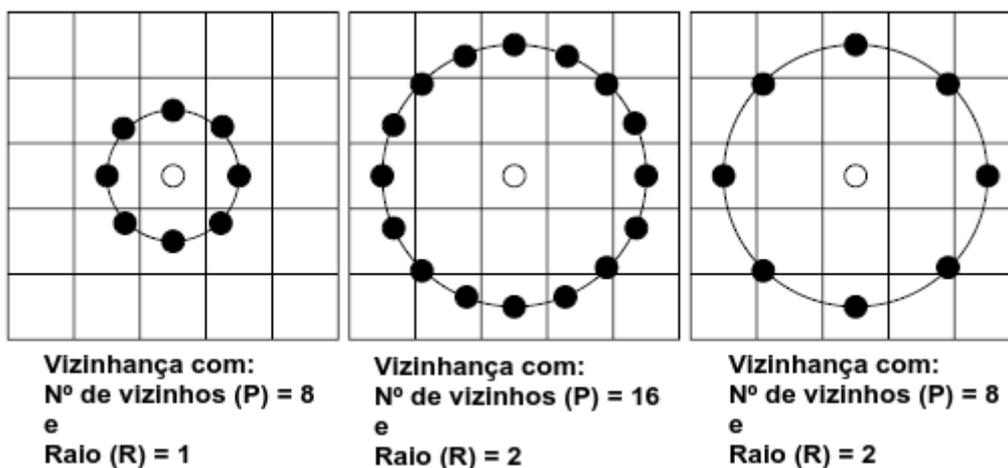


Figura 3.5: Vizinhanças circulares (Fonte: Adaptado de (PIETIKÄINEN et al., 2011, p. 14))

Com base nesse esquema, considere que:

$$g_p = I(x_p, y_p), \quad \text{com } p \text{ variando de } 0 \text{ at } P - 1 \text{ e} \quad (3.1)$$

$$x_p = x + R \cos(2\pi p/P), \quad (3.2)$$

$$y_p = y + R \sin(2\pi p/P). \quad (3.3)$$

Considere que a textura local da imagem no ponto $I(x, y)$ é caracterizada pela distribuição conjunta dos valores de cinza de $P + 1$ *pixels* (o *pixel* central e seus P vizinhos, considerando $P > 0$):

$$T = t(g_c, g_0, g_1, \dots, g_{p-1}) \quad (3.4)$$

Sem perda de informação, o valor do *pixel* central pode ser subtraído da vizinhança:

$$T = t(g_c, g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c) \quad (3.5)$$

Em seguida, a distribuição conjunta é aproximada, por entender que o *pixel* central é estatisticamente independente das diferenças. Esse entendimento viabiliza a fatorização (decomposição de um objeto no produto de outros objetos) da distribuição:

$$T \approx t(g_c) t(g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c) \quad (3.6)$$

Nesse momento, o primeiro fator - $t(g_c)$ - representa a distribuição de intensidade sobre $I(x, y)$, e, sob o prisma da análise de padrões de textura local, ele não contém nenhuma informação útil. De maneira contrária, sob o mesmo prisma, o segundo fator - $t(g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c)$ - pode ser utilizado para modelar a textura local. Sobretudo, a estimação dependente dessa distribuição multidimensional de dados da imagem pode ser difícil.

Com o objetivo de atenuar esses desafios, para o segundo fator, resultante da fatorização (que será utilizado para modelar a textura local) apenas os sinais das diferenças são considerados, ou seja, o referido fator passa a ser utilizado da seguinte maneira:

$$t(s(g_0 - g_c), s(g_1 - g_c), \dots, s(g_{p-1} - g_c)) \quad (3.7)$$

Entendendo que $s(x)$ é uma função de limiar (*thresholding*), que tem o seguinte comportamento:

$$s(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (3.8)$$

Como acontece no caso do *LBP* básico, este *LBP* será obtido pelo somatório dos resultados da função de limiar, multiplicado pelo potência $2^{P(\text{Num. de vizinhos})}$. Assim, o operador $LBP_{P,R}$ é definido como:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.9)$$

Na prática, isso significa que os limiares das diferenças, obtidos à partir da vizinhança considerada, formarão um número binário de P bits. De onde se conclui que o código *LBP* resultante possui um domínio (valores possíveis) bem definido, com 2^P valores possíveis. Por fim, a distribuição de escala de cinza local (a textura) pode ser descrita com, aproximadamente, uma distribuição discreta de 2^P códigos *LBP*:

$$T \approx t(LBP_{P,R}(x_c, y_c)) \quad (3.10)$$

Cálculo dos histogramas

Os histogramas são calculados após a aplicação do operador $LBP_{P,R}$. O histograma final (que é o resultado da concatenação dos histogramas correspondentes a várias seções da imagem) pode ser utilizado para representar a própria imagem, no método que está sendo proposto. Observe a Figura 3.6, ela ilustra o cálculo de um histograma.

Neste contexto, considerando que cada imagem é representada por um histograma, a Base de Conhecimento, construída durante a *Etapa 2*, é formada por estes histogramas e pelas classes das respectivas imagens. Analogamente, durante a fase preditiva (*Etapa 3*) outros histogramas representaram as informações a serem classificadas.

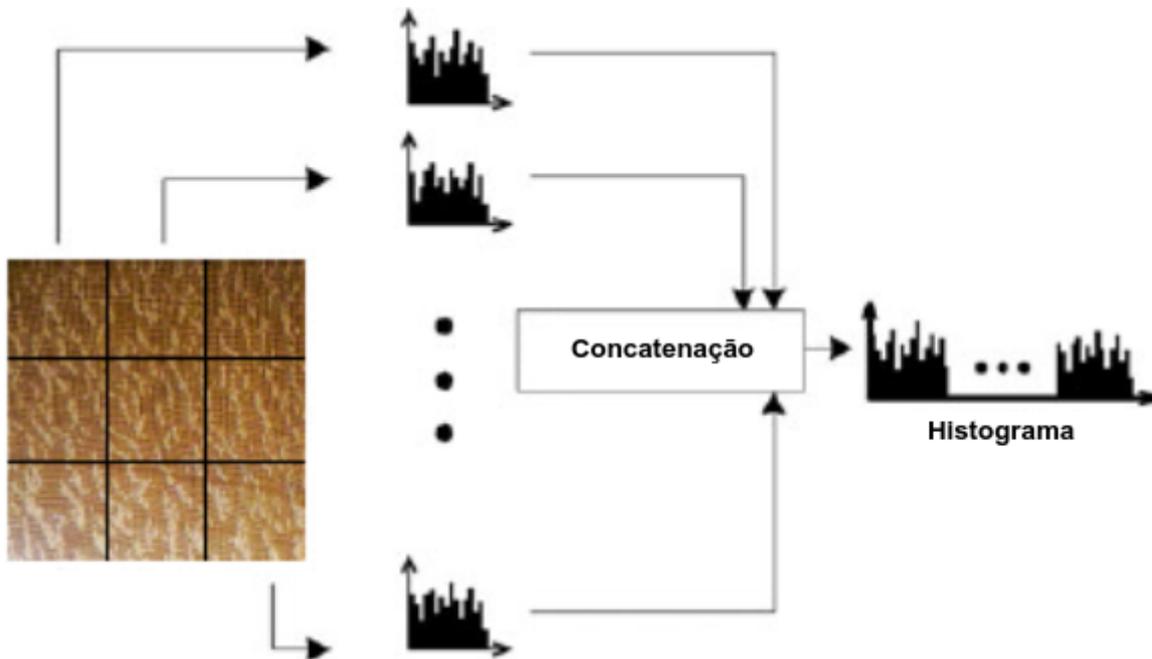


Figura 3.6: Histograma calculado à partir de uma amostra

Normalização

Dentro do processo de Extração de características, o último processo interno diz respeito a uma etapa de preparação dos histogramas. Esse procedimento de preparação é um processamento matemático conhecido como *normalização* (explorado na Seção 2.1.7). O método proposto trabalhará com a normalização do tipo *Norma de Um*.

O objetivo deste processo é majorar os resultados alcançados pelo método, com base em uma comparação mais justa entre os atributos das instâncias em análise.

Conforme se vê no Quadro 2.1, entenda que cada registro, na Base de Conhecimento, pode ser referenciado como uma instância. E esse registro possui algumas propriedades, que serão referenciadas, no método proposto, como atributos.

Durante o processo preditivo (realizado na *Etapa 3*) os atributos, das instâncias da Base de Conhecimento e das instâncias submetidas para avaliação, são utilizados como insumos para viabilizar a predição.

Para complementar o entendimento referente aos efeitos causados nos histogramas, em função do processo de normalização, observe a Figura 3.7.

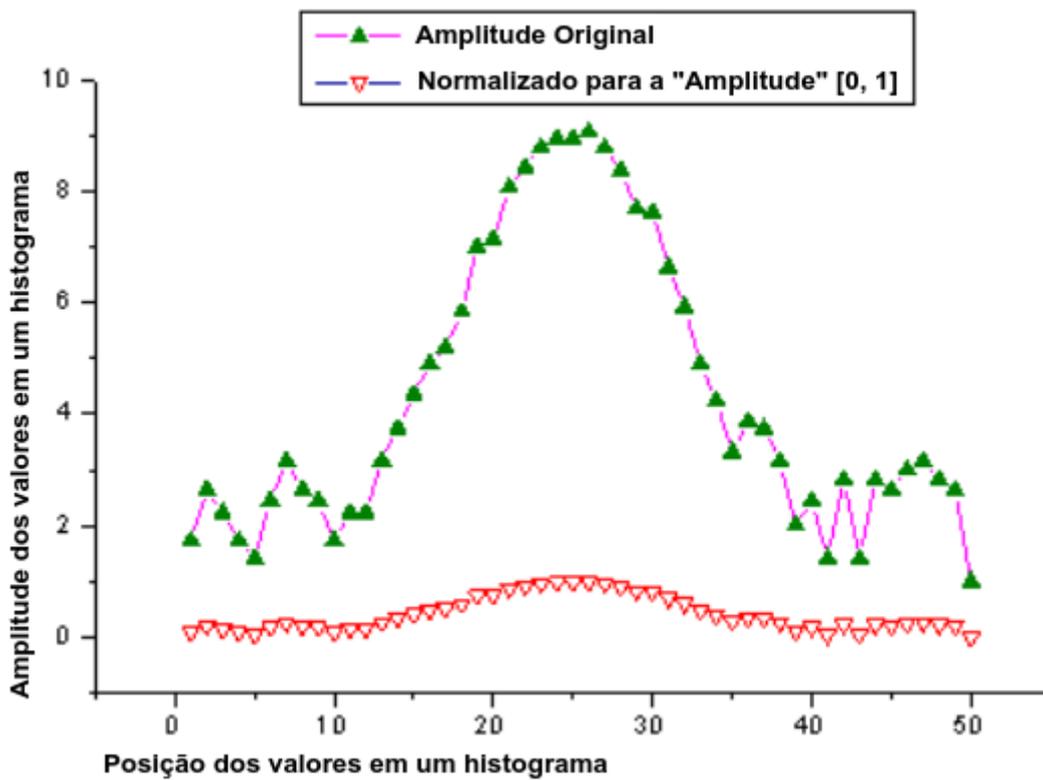


Figura 3.7: Efeitos da Normalização

Como é possível observar na Figura 3.7, existem duas linhas bem distintas: a primeira (com triângulos verdes) possui uma amplitude bem maior, e representa os valores alocados pelo histograma antes do processo de normalização; a segunda (com triângulos vermelhos) possui uma amplitude bem menor, em função do resultado da normalização, que atenua as diferenças de amplitude percebidas no conjunto.

3.2 Etapa 1 - Construção do banco de dados das imagens

O objetivo desta etapa é construir o Banco de Dados das Imagens, que serve de insumo para a construção da Base de Conhecimento, no escopo do método proposto. A Figura 3.8 mostra o fluxo dos processos dessa etapa.

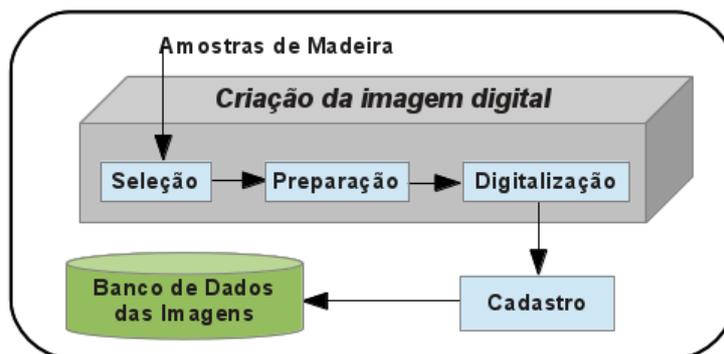


Figura 3.8: Etapa 1 - Construção do banco de dados das imagens

Todos os processos internos do processo de Criação da imagem digital, que existe na *Etapa 1* e na *Etapa 3*, já foram explanados na Seção 3.1.1. E, quando todos esses processos internos, do processo Criação da imagem digital, já tiverem sido executados, o único processo pendente, para finalizar o fluxo da *Etapa 1* do método proposto, é o processo de *cadastro*, que consiste no armazenamento das imagens digitais dentro do Banco de Dados das Imagens.

O conteúdo do Banco de Dados das Imagens, que é o produto final desta etapa, pode ser utilizado integralmente, ou parcialmente, para a construção da Base de Conhecimento do método proposto, isso é decidido durante a *Etapa 2*.

O universo de possibilidades para a construção da Base de Conhecimento, do método proposto, é o Banco de Dados das Imagens: as características extraídas, das imagens que formam o Banco de Dados das Imagens, são utilizadas para compor a Base de Conhecimento, durante a *Etapa 2* do método em questão.

Desta maneira, quando todas as características das imagens, do Banco de Dados das Imagens, já estiverem cadastradas na Base de Conhecimento, será necessário realizar novos cadastros no Banco de Dados das Imagens, caso exista o desejo de ampliar a Base de Conhecimento, nos moldes do método proposto.

3.3 Etapa 2 - Construção da base de conhecimento

O objetivo da *Etapa 2* é a construção da Base de Conhecimento. O fluxo desta etapa pode ser visto na Figura 3.9.

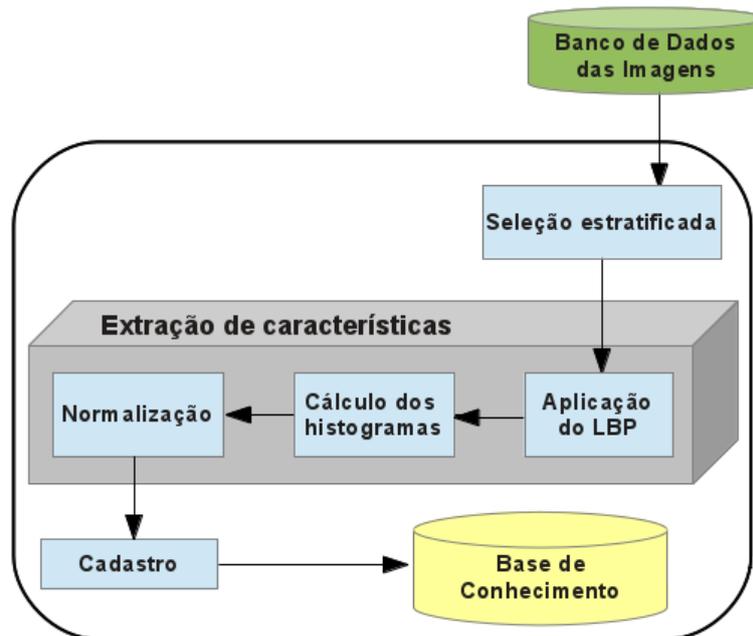


Figura 3.9: Etapa 2 - Construção da base de conhecimento

A construção da Base de Conhecimento se inicia com a Seleção estratificada das imagens, que já existem no Banco de Dados das Imagens, construído durante a *Etapa 1*.

Segundo Refaeilzadeh, Tang e Liu (2008, p. 2), estratificação é o processo de rearranjar os dados, para garantir que cada classe tenha uma boa representatividade no todo: por exemplo, em um problema de classificação binária, no qual o classificador possui um universo de duas respostas possíveis, o ideal é que a base de dados, utilizada para treinamento, possua, mais ou menos, metade dos registros representando cada tipo de resposta (classe) possível. Desta forma, em função da Seleção estratificada, o classificador será treinado com intensidade semelhante para cada resposta (classe), dentro do seu universo de respostas possíveis.

Todos os processos internos, do processo de Extração de características, existente na *Etapa 2* e na *Etapa 3*, já foram explanados na Seção 3.1.2.

Quando a Seleção estratificada, e todos os processos internos da Extração de características, já tiverem sido executados, o único processo pendente, para finalizar o fluxo da *Etapa 2*, no

método proposto, é o processo de *cadastro*, que consiste no armazenamento, dentro da Base de Conhecimento, de um registro com duas informações:

- O histograma normalizado (produto final do processo de Extração de características);
- A classe (espécie) à qual a amostra, que deu origem à imagem, pertence.

A Base de Conhecimento é o universo à partir do qual o classificador faz suas inferências, desta forma, qualquer tentativa de predição, acerca de classes (espécies) inexistentes na Base de Conhecimento, não retornará resultados coerentes.

3.4 Etapa 3 - Realizando predições

Na *Etapa 3* é realizada a *predição*, conforme fluxo visto na Figura 3.10.

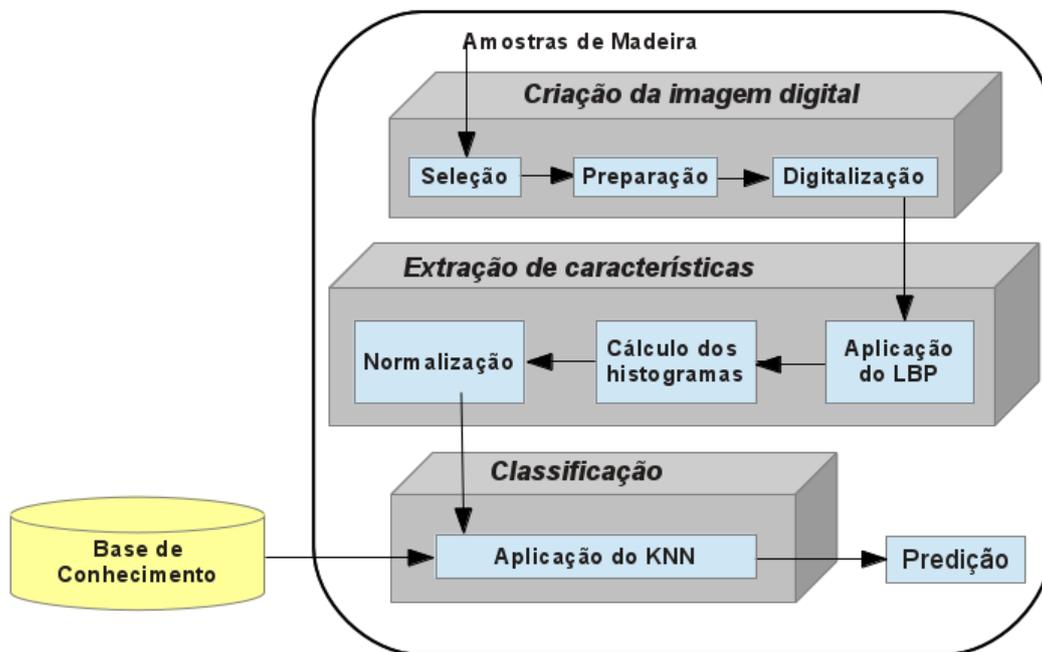


Figura 3.10: Etapa 3 - Realizando predições

A *predição*, no escopo do método proposto, é o diagnóstico acerca da classe, da amostra de madeira, que deu origem à imagem submetida para avaliação. A Base de Conhecimento (produto final da *Etapa 2*) serve de subsídio para a *predição*, que é realizada através do algoritmo *K-NN* (explorado na Seção 2.1).

Nesta etapa, as amostras, selecionadas para o processo preditivo, precisam pertencer ao universo de classes existentes na Base de Conhecimento (que são frutos da Seleção estratificada realizada durante a *Etapa 2*), pois, o classificador só deve fazer predições para o universo de classes para as quais foi treinado.

Todos os processos internos dos processos comuns (*criação da imagem digital* e Extração de características) já foram explanados na Seção 3.1.2.

Quando todos os processos internos da *criação da imagem digital* já tiverem sido executados, e todos os processos internos da Extração de características também já tiverem sido executados, o único processo pendente, para finalizar o fluxo da *Etapa 3*, no método proposto, é o processo de *classificação*, que disponibiliza, como produto final, a *predição*.

A *classificação* é o processo final da *Etapa 3*, conforme visto na Figura 3.10, e é realizada utilizando um algoritmo baseado em instância (vide seção 2.1.8) conhecido como *K-NN*. Esse algoritmo (*K-NN*) já foi detalhado na Seção 2.1.

A Figura 3.11 ilustra, apontando apenas alguns processos existentes na *Etapa 3*, como o método proposto chega à predição.

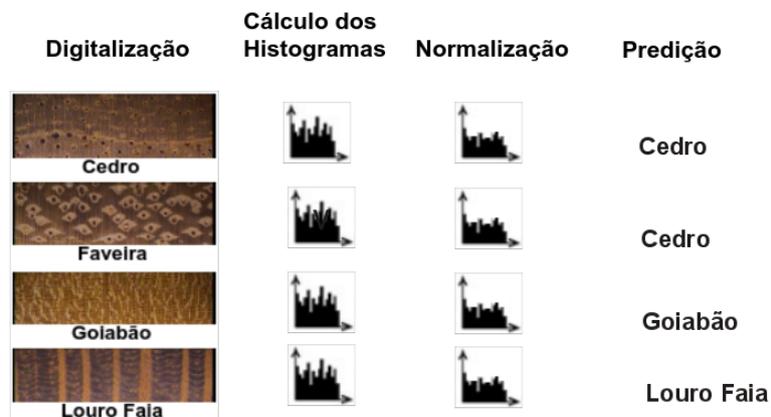


Figura 3.11: Passos da Etapa 3

CAPÍTULO 4

EXPERIMENTOS

Este capítulo aborda os resultados dos experimentos, com as respectivas estatísticas e análises. É detalhado como o banco de dados das imagens foi construído. São abordadas todas as ferramentas utilizadas. E, após relatar o resultado de cada experimento específico, é realizada uma análise geral acerca dos efeitos da normalização, das técnicas de *Cross-Validation* e das métricas de distância.

4.1 Banco de dados das imagens

As imagens que formaram o banco de dados utilizado nesses experimentos foram concedidas por duas empresas: a *Embrapa*, que forneceu 63 imagens, e o *VRI(UFPR)*, que forneceu 2240 imagens, totalizando 2303 imagens de 121 espécies. Cada espécie foi organizada dentro de um grupo, como pode ser visto na Tabela abaixo.

Tabela 4.1: Resumo quantitativo das imagens desta pesquisa

Grupo	Fonte	Classe	Quantidade de Imagens
1	Embrapa	Andiroba	10
2	Embrapa	Cedro	19
3	Embrapa	Faveira	04
4	Embrapa	Goiabão	04
5	Embrapa	Guariuba	04
6	Embrapa	Louro Faia	04
7	Embrapa	Mogno	10
8	Embrapa	Tauari	04
9	Embrapa	Taxirana	04
10...121	UFPR	...	20 para cada Grupo (Tipo de Imagem)
			TOTAL = 2303 imagens

4.1.1 Ferramentas utilizadas

As amostras de madeira, selecionadas para as fotografias, passaram por procedimentos padronizados de tratamento e armazenamento, conforme organização imposta pela xiloteca: As superfícies fotografadas, em todas as amostras, sofreram cortes transversais (Fonte: (EMBRAPA, 1972)).

As imagens da Embrapa foram coletadas diretamente da xiloteca, que existe na unidade referenciada como *Embrapa Amazônia Oriental*.

Para o preparo das superfícies (polimento), foi utilizado um micrótomo, visto na Figura 3.2a (Fonte: (EMBRAPA, 1972)).

Para coletar as imagens foi utilizada uma câmera apropriada para micro fotografias, vista na Figura 4.1a, que foi acoplada em uma lupa, vista na Figura 4.1b. A câmera utilizada pode ser acoplada nas oculares de qualquer marca de microscópio ou estereoscópio (Fonte: (EMBRAPA, 1972)).



(a) Câmera



(b) Câmera acoplada na lupa

Figura 4.1: Câmera utilizada (Fonte: (EMBRAPA, 1972))

Além dos equipamentos mencionados, outros tipos de lupa foram utilizadas para interagir com as amostras da xiloteca. A Figura 4.2a mostra uma lupa que fornece um aumento de 10 vezes, e que não possui luminária. Já a Figura 4.2b mostra um funcionário da *Embrapa* interagindo com uma lupa que também possui aumento de 10 vezes, e disponibiliza uma luminária para auxiliar nas observações.

Adicionalmente, o *VRI(UFPR)* concedeu o acesso ao *Forest Species Database*, que pode



(a) Lupa sem luminária



(b) Lupa com luminária

Figura 4.2: Outras lupas (Fonte: (EMBRAPA, 1972))

ser acessado através do endereço eletrônico <http://web.inf.ufpr.br/vri/image-and-videos-databases/forest-species-database>. Esta base é composta por 2240 imagens macroscópicas de 112 espécies diferentes de madeiras, pertencentes a dois grupos específicos: madeiras *Hardwood* - resistentes, utilizadas para produzir mobília; e madeiras *Softwood* - mais flexíveis, fáceis de cortar, e inapropriadas para produção de mobília.

As imagens deste banco (*Forest Species Database*) foram capturadas fazendo uso de um microscópio, considerando um zoom de 100 vezes. As imagens resultantes foram salvas no formato PNG (*Portable Network Graphics*) sem compressão, e com uma resolução de 1027 x 768 pixels, conforme descrição encontrada no site mencionado.

Todo o trabalho de codificação do método, desenvolvido neste projeto, foi feito utilizando o *Matlab*, que é um software desenvolvido pela empresa *MathWorks* (outras informações, acerca desta empresa, podem ser vistas em <http://www.mathworks.com>).

4.1.2 Criação da imagem digital

O processo de criação das imagens foi totalmente executado pela *Embrapa* e pelo *VRI(UFPR)*, através dos meios descritos na Seção 4.1.1. Para a realização desta tarefa, eles estabeleceram seus próprios critérios. Conforme o método proposto, esse processo possui três processos internos:

Seleção

A seleção de amostras de madeira, realizada pela *Embrapa*, foi feita com base nas classes de madeira sob as quais incidem os maiores índices de tráfico ilegal (Fonte: (EMBRAPA, 1972)). Quanto às imagens disponibilizadas pelo *VRI(UFPR)*, não foram encontradas informações, na página do referido banco de imagens, acerca dos critérios utilizados para a seleção das amostras de madeira.

Preparação

Segundo informações da *Embrapa*, o preparo diz respeito a um polimento realizado nas superfícies que serão fotografadas. O objetivo deste polimento é majorar a qualidade das imagens que serão capturadas, e ele é realizado através do micrótomo, visto na Figura 3.2a.

Após o polimento de cada superfície, lupas são utilizadas para verificar se a qualidade almejada já foi atingida, ou se ainda é necessário repetir o processo com essa finalidade. As lupas, utilizadas neste processo de verificação, podem ser vistas nas Figuras 4.2a e 4.2b.

Adicionalmente, todas as amostras fornecidas pelo *VRI(UFPR)* (2240 imagens) parecem ter um padrão similar de qualidade, o que pode ser um indício alusivo à existência de uma etapa de preparação da matéria prima, em moldes semelhantes aos que foram detalhados pela *Embrapa*.

Digitalização

Cada amostra de madeira (*corpo de prova*) foi preparada de modo que suas faces transversais recebessem um acabamento superficial (polimento), exatamente no local à partir do qual as imagens seriam capturadas. Essa abordagem aumenta a qualidade das imagens. Para cada espécie disponibilizada, foram coletadas amostras de três árvores distintas (Fonte: (EMBRAPA, 1972)).

Quanto ao material disponibilizado pelo *VRI(UFPR)* (a base de dados *Forest Species Database*) é possível afirmar que as informações relacionadas à captura das imagens, que formam esta base, estão publicamente disponíveis na Internet, através da página que descreve tal base. Nesta página é possível encontrar informações acerca do *hardware* utilizado na captura das imagens.

Nem todas as imagens coletadas são selecionadas para o *banco de dados das imagens*. Após o término da *digitalização*, as imagens resultantes serão analisadas, objetivando eleger as melhores para inclusão no *banco de dados das imagens*. As imagens que não são selecionadas para o *banco de dados das imagens* são descartadas.

Segundo a (EMBRAPA, 1972), o critério considerado para a adoção, ou descarte, de cada imagem, foi a nitidez (sob o prisma da observação humana), que é uma função da qualidade do corte transversal feito com o micrótomo.

Considerando a base de dados disponibilizada pelo VRI(UFPR) (a base *Forest Species Database*), não estão descritos, no endereço eletrônico da referida base, os detalhes considerados para que uma imagem seja selecionada, ou descartada, após a sua *digitalização*.

4.2 Resultados

Nesta seção do trabalho serão encontrados os resultados dos experimentos realizados, e as respectivas análises para cada experimento específico.

4.2.1 Teste com parte das imagens da Embrapa e sem usar Cross-Validation

Neste trabalho estão sendo considerados vários tipos de madeira (121 espécies), entretanto, na primeira bateria de testes, estavam disponíveis poucas imagens (apenas 28), referentes às espécies: *Cedro*, *Faveira*, *Goiabão*, *Guariuba*, *Louro Faia*, *Tauari* e *Taxirana*. Nesse teste inicial, para cada uma das madeiras citadas, foram utilizadas apenas quatro imagens. Todas as imagens utilizadas neste teste estão listadas no Quadro 4.1, e os resultados podem ser observados na Tabela 4.2.

Nesses primeiros testes, utilizando as imagens dispostas no Quadro 4.1, para cada tipo de madeira, apenas uma imagem foi utilizada para treinamento, e todas as outras foram utilizadas para teste. E, mesmo com uma quantidade muito pequena de imagens de treinamento (apenas uma imagem por espécie), o método já obteve os resultados apresentados na Tabela 4.2.

Quadro 4.1: Amostras utilizadas para a primeira bateria de testes

Madeira	Amostras
Cedro	cedro1; cedro2; cedro3; cedro4;
Faveira	faveira1; faveira2; faveira3; faveira4;
Goiabão	goiabao1; goiabao2; goiabao3; goiabao4;
Guariuba	guariuba1; guariuba2; guariuba3; guariuba4;
Louro Faia	lourofaia1; lourofaia2; lourofaia3; lourofaia4;
Tuari	tauari1; tauari2; tauari3; tauari4;
Taxirana	taxirana1; taxirana2; taxirana3; taxirana4;

Fonte: (EMBRAPA, 1972)

Tabela 4.2: Resultado da primeira bateria de testes

Madeira	Acerto (%)
Cedro	33,33
Faveira	33,33
Goiabão	100
Guariuba	100
Louro Faia	100
Tuari	100
Taxirana	100

Nesses testes foi utilizado o algoritmo *K-NN*. O *K* foi igual a 1, e a distância utilizada foi a *Euclidiana*.

A primeira bateria de testes foi uma experiência muito limitada: poucas imagens foram usadas, e nenhuma técnica de *Cross-Validation* foi utilizada. Mesmo assim, o método já obteve, nesses primeiros testes, aproximadamente, 81% de acerto (mensurado pela média aritmética dos percentuais de acerto da Tabela 4.2).

Esses testes representam a primeira experiência de categorização com as imagens recebidas da *Embrapa*. Foi possível fazer uma inferência nesta primeira análise: sob a ótica do olho humano, as duas categorias de madeira que apresentaram erros (Cedro e Faveira) eram visualmente parecidas entre si, conforme se vê na Figura 4.3.

E essas duas categorias de madeira (Cedro e Faveira) eram muito diferentes de todas as outras. Para justificar essa afirmação, visualmente, foram selecionadas duas categorias (Goiabão e Louro Faia) para que seja possível comparar com as categorias em questão (Cedro e Faveira), através da Figura 4.4:

Analisando as imagens, sob a perspectiva do olho humano, embora seja possível diferenciar



(a) Imagem do Cedro



(b) Imagem da Faveira

Figura 4.3: Imagens de Cedro e Faveira (Fonte: (EMBRAPA, 1972))



(a) Imagem do Goiabão



(b) Imagem de Louro Faia

Figura 4.4: Imagens de Goiabão e Louro Faia (Fonte: (EMBRAPA, 1972))

Cedro de Faveira, dentro do contexto das imagens desta bateria de testes, provavelmente, um olho humano só confundiria Cedro com Faveira, e não confundiria com nenhuma das demais classes no referido contexto, pois, essas duas classes de imagens são muito diferentes de todas as outras, como é possível perceber analisando as Figuras 4.3 e 4.4.

Adicionalmente, analisando os resultados deste teste, foi possível perceber que sempre que a categorização do Cedro falhava, o método imaginava que se tratava da Faveira, e, analogamente, sempre que a categorização da Faveira falhava, o método imaginava que se tratava do Cedro. À partir desta situação, foi possível inferir que, para este universo de classes, o que era mais parecido para o olho humano, também foi mais parecido para o método.

4.2.2 Teste com todas as imagens da Embrapa, usando Cross-Validation

Nesta bateria de testes, estavam sendo utilizadas as mesmas amostras do Quadro 4.1 acrescidas de outras 35 imagens disponibilizadas pela *Embrapa*, de maneira que, neste teste, todas as imagens disponibilizadas pela *Embrapa* já estavam sendo utilizadas. Essas imagens podem ser observadas na Tabela 4.3.

Tabela 4.3: Todas as imagens fornecidas pela *Embrapa*

Grupo	Fonte	Classe	Quantidade
1	Embrapa	Andiroba	10
2	Embrapa	Cedro	19
3	Embrapa	Faveira	04
4	Embrapa	Goiabão	04
5	Embrapa	Guariuba	04
6	Embrapa	Louro Faia	04
7	Embrapa	Mogno	10
8	Embrapa	Tauari	04
9	Embrapa	Taxirana	04
TOTAL =			63 imagens

Na segunda bateria de testes foram utilizadas 63 imagens (todas as imagens fornecidas pela *Embrapa*). E, independente da abordagem de *Cross-Validation* adotada, ou de qualquer outro ajuste, o método sempre retornava 100% de acerto.

A partir desta situação foi possível inferir que:

- A primeira bateria de testes atingiu apenas 81% de acerto, mesmo com uma quantidade de imagens bem inferior (apenas 28 imagens), porque não considerou as técnicas de *Cross-Validation*;
- As imagens disponíveis, neste momento da pesquisa, eram insuficientes para estudar a viabilidade dos métodos adotados: Existiam poucas classes de imagem, e elas ainda possuíam padrões bem distintos, mesmo para um olho humano;
- Mesmo as classes mais parecidas deste conjunto (Cedro e Faveira - conforme explanado na análise da primeira bateria de testes), com a incorporação das metodologias de *Cross-Validation*, não se confundiam mais, de maneira contrária ao que ocorria na primeira bateria de testes;
- Foi necessário pensar numa alternativa para que novos testes pudessem ser realizados, enquanto novas imagens não eram disponibilizadas. Essa reflexão levou à postura adotada na bateria de testes que será abordada em seguida.

4.2.3 Teste com todas as imagens da Embrapa, adicionando ruído

Essa bateria de testes foi motivada pela percepção da necessidade de novas imagens, em função de uma base de imagens muito pequena, com a qual vários testes já tinham sido realizados. No momento da execução desta bateria de testes, estavam disponíveis, apenas, as imagens observadas na Tabela 4.3, desta forma, o número de imagens ainda era muito pequeno e o método continuava exibindo 100% de acerto nos testes.

A abordagem utilizada para viabilizar novos testes, enquanto novas imagens não eram disponibilizadas, foi a geração de cópias das imagens disponíveis, com diferentes níveis de ruído, com o intuito de observar o poder de resposta, dos algoritmos adotados, neste novo cenário.

Em todos os testes, descritos nesta seção, estão sendo consideradas as imagens fornecidas pela *Embrapa* (observadas na Tabela 4.3). Também está sendo considerada, para geração de todas as amostras ruidosas, a função *imnoise* do *Matlab*, parametrizando a opção de algoritmo *sal e pimenta*, com diferentes níveis de ruído.

A terceira bateria de testes foi estruturada em três etapas:

- **Primeira etapa:** Considerando a inserção de ruído apenas nas instâncias de validação, mantendo as instâncias de treinamento sem ruído;
- **Segunda etapa:** Inserindo ruído em todas as instâncias utilizadas (tanto nas de treinamento quanto nas de validação);
- **Terceira etapa:** Inserindo ruído em um percentual das instâncias utilizadas no teste;

Os resultados dessa primeira etapa, na qual apenas as instâncias de validação foram modificadas com a inserção de ruído, mostram a sensibilidade do método em relação a um contexto ruidoso, e podem ser vistos nas Tabelas 4.4 e 4.5.

Tabela 4.4: Etapa 1 - Ruído no conjunto de validação (*K-Fold Cross-Validation*)

Ruído (%)	Média de Acerto (%)	Desvio Padrão	Coef. de Variação (%)
1	30,16	2,82	9.37
2	30,16	2,82	9.37
5	22,05	14,47	65.62

A abordagem de não inserir ruído no conjunto de treinamento, e inserir ruído nas imagens utilizadas para validação, mostrou o nível de impacto que o ruído pode causar nos resultados do

Tabela 4.5: Etapa 1 - Ruído no conjunto de validação (*Leave-One-Out Cross-Validation*)

Ruído (%)	Acerto (%)
1	34,92
2	31,75
5	25,40

algoritmo *K-NN*. Basta recordar que a categorização, através do algoritmo *K-NN*, deste mesmo conjunto de amostras, sem inserção de ruído, com qualquer abordagem de *Cross-Validation*, estava resultando em 100% de acerto, e, após a inserção de 5% de ruído nas amostras utilizadas para validação, o método só conseguiu atingir 22,05% de acerto com a abordagem *K-Fold Cross-Validation* e 25,40% de acerto com a abordagem *Leave-One-Out Cross-Validation*.

Conforme explanado na Seção 2.1, a categorização é realizada pela mensuração de distâncias, entre a amostra a ser categorizada e as amostras do conjunto de treinamento, em um espaço *n*-dimensional. Quando foi adotada a postura de inserir ruído apenas nas amostras utilizadas para validação, o espaço entre essas amostras, e as amostras do conjunto de treinamento, aumentou, e, como consequência, os resultados do *K-NN* ficaram piores. Observe o Gráfico 4.1 construído com os dados das Tabelas 4.4 e 4.5.

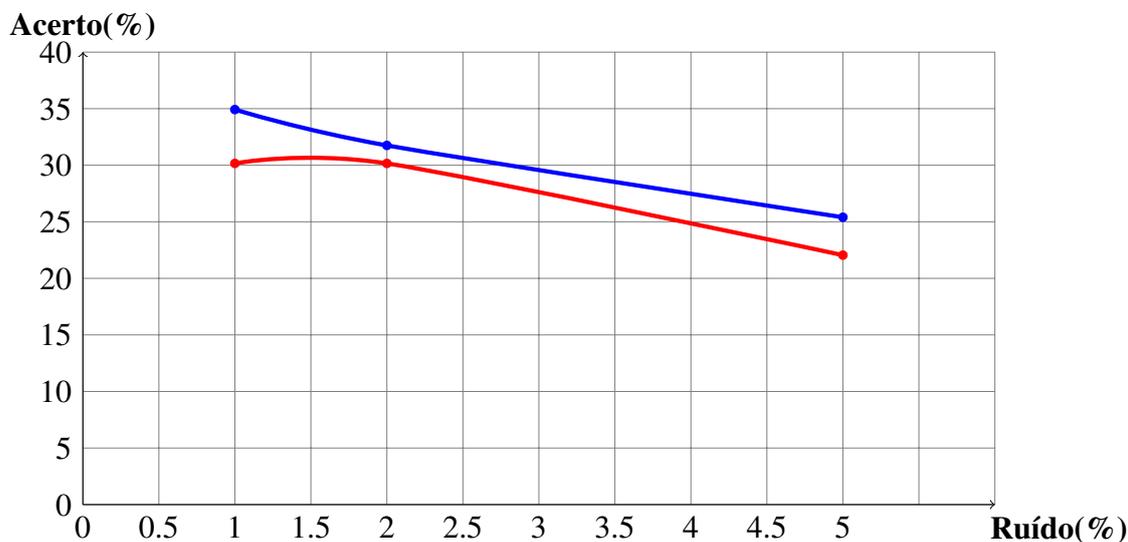


Gráfico 4.1: Teste inserindo ruído apenas no conjunto de validação

Legenda do Gráfico 4.1:

Linha 1 (—): Ruído nas instâncias de validação; Abordagem 4FCV

Linha 2 (—): Ruído nas instâncias de validação; Abordagem LOOCV

Esses resultados motivaram a inserção de ruído em todas as amostras (nas do conjunto de treinamento e nas do conjunto de validação). Essa postura foi adotada na segunda etapa desta mesma bateria de testes. O objetivo era investigar se essa abordagem culminaria em resultados melhores que a abordagem utilizada na primeira etapa (inserir ruído apenas no conjunto de validação). Os resultados da segunda etapa, desta bateria de testes, podem ser verificados nas Tabelas 4.6 e 4.7.

Tabela 4.6: Etapa 2 - Ruído em todas instâncias (*K-Fold Cross-Validation*)

Ruído (%)	Média de Acerto (%)	Desvio Padrão	Coef. de Variação (%)
5	88,43	14,82	16,76
15	86,96	13,96	16,06
25	76,04	13,32	17,52
35	72,77	10,38	14,26
45	71,01	15,94	22,45
55	72,68	13,66	18,80
65	64,56	17,35	26,87
75	55,14	13,13	23,82
85	47,32	13,68	28,91
95	37,62	15,64	41,58
100	31,63	8,54	27,00

Tabela 4.7: Etapa 2 - Ruído em todas instâncias (*Leave-One-Out Cross-Validation*)

Ruído (%)	Acerto (%)
5	92,06
15	92,06
25	82,54
35	76,19
45	74,60
55	77,78
65	69,84
75	61,90
85	53,97
95	49,21
100	28,57

Sob o prisma da observação da resistência ao ruído, essa abordagem (de inserir ruído tanto nas instâncias de treinamento quanto nas de validação) já mostrou resultados bem mais tolerantes que a abordagem anterior (de inserir ruído apenas nas amostras de validação), conforme pode ser verificado nas Tabelas 4.6 e 4.7. Observe também o Gráfico 4.2 que é construído a partir dos dados destas Tabelas.

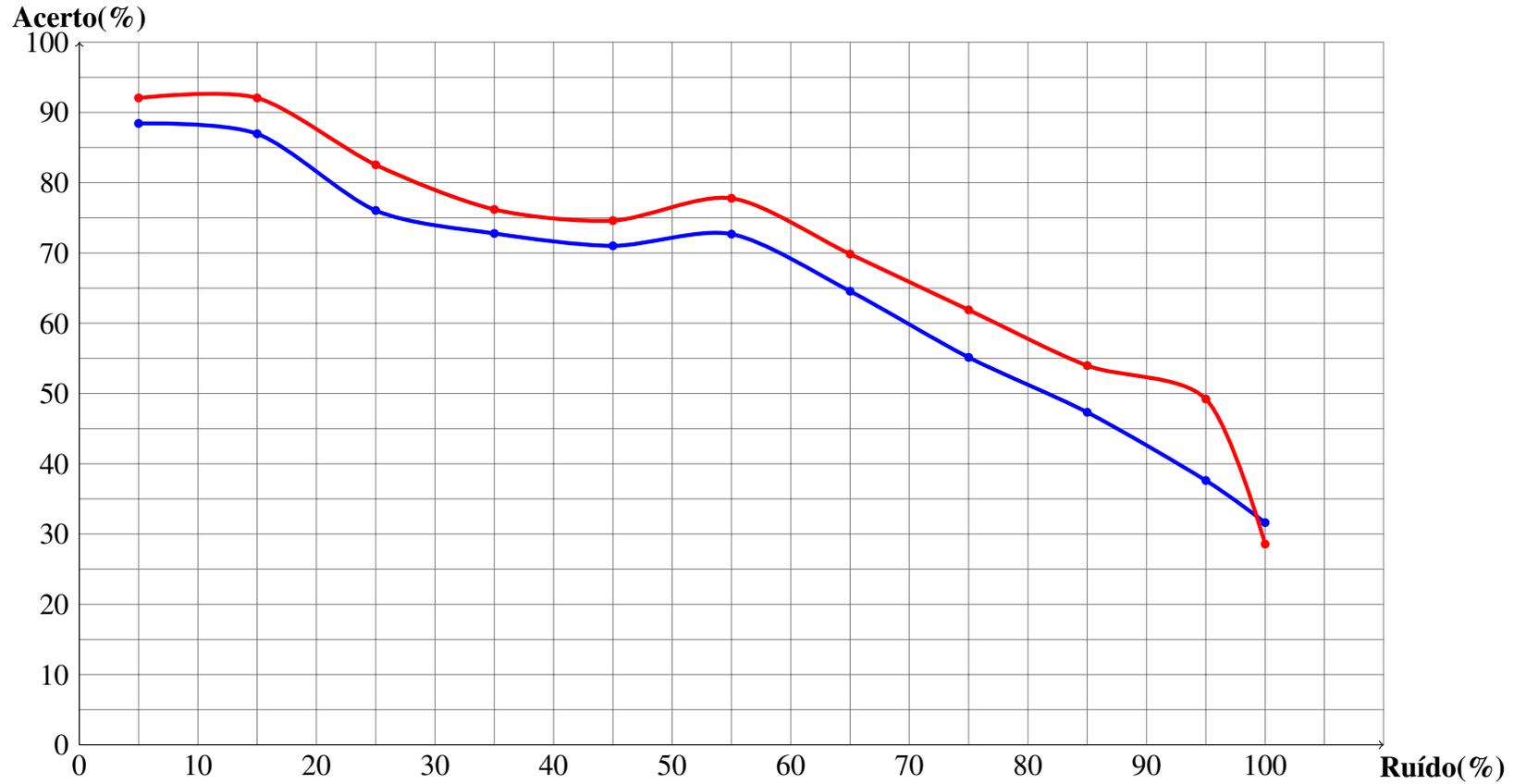


Gráfico 4.2: Teste inserindo ruído em todas as instâncias

Legenda do Gráfico 4.2:

Linha 1 (—): Ruído em todas as instâncias; Abordagem 4FCV

Linha 2 (—): Ruído em todas as instâncias; Abordagem LOOCV

Nos testes realizados na primeira etapa desta bateria, o ruído esteve presente apenas nas instâncias de validação. Nos testes realizados na segunda etapa, o ruído esteve presente em todas as instâncias. De qualquer forma, parece mais provável que, em situações reais, apenas uma parte das amostras (seja no conjunto de treinamento, no de validação ou em ambos) possua ruído. Com base nesse raciocínio, surgiu a motivação de realizar um último teste acerca da resistência ao ruído: degradando apenas um percentual das amostras que serão utilizadas, alternadamente, tanto para treinamento quanto para validação. Esse teste pode ser observado nas Tabelas 4.8 e 4.9, e, também, nos Gráficos que são construídos à partir de tais Tabelas (Gráficos 4.3 e 4.4).

Tabela 4.8: Etapa 3 - Ruído em parte instâncias (*K-Fold Cross-Validation*)

Ruído (%)	Amostras Corrompidas (%)	Média (%)	Desvio Padrão	Coef. de Variação (%)
10	15.87	90.00	12.76	14.18
10	46.03	93.33	13.33	14.28
10	52.38	93.33	13.33	14.28
10	58.73	93.33	13.33	14.28
10	65.07	93.33	13.33	14.28
10	71.42	93.33	13.33	14.28
10	87.30	93.33	13.33	14.28
10	93.65	91.66	16.65	18.18
10	100	88.53	14.87	16.80
35	15.87	90.00	12.76	14.18
35	46.03	90.19	11.63	12.89
35	52.38	86.96	9.71	11.17
35	58.73	86.96	9.71	11.17
35	65.07	83.83	11.83	14.12
35	71.42	83.83	11.83	14.12
35	87.30	74.24	12.41	16.71
35	93.65	74.24	12.41	16.71
35	100	72.77	10.38	14.26
65	15.87	90.00	12.76	14.18
65	46.03	90.20	12.70	14.08
65	52.38	85.50	10.02	11.73
65	58.73	83.84	8.77	10.46
65	65.07	80.51	14.65	18.20
65	71.42	75.81	11.86	15.65
65	87.30	71.01	12.85	18.10
65	93.65	66.12	15.81	23.92
65	100	64.56	17.35	26.87

Esses dados serão vistos de forma mais intuitiva através de um gráfico, com base nisso, o Gráfico 4.3 foi construído à partir dos dados dessa Tabela (Tabela 4.8).

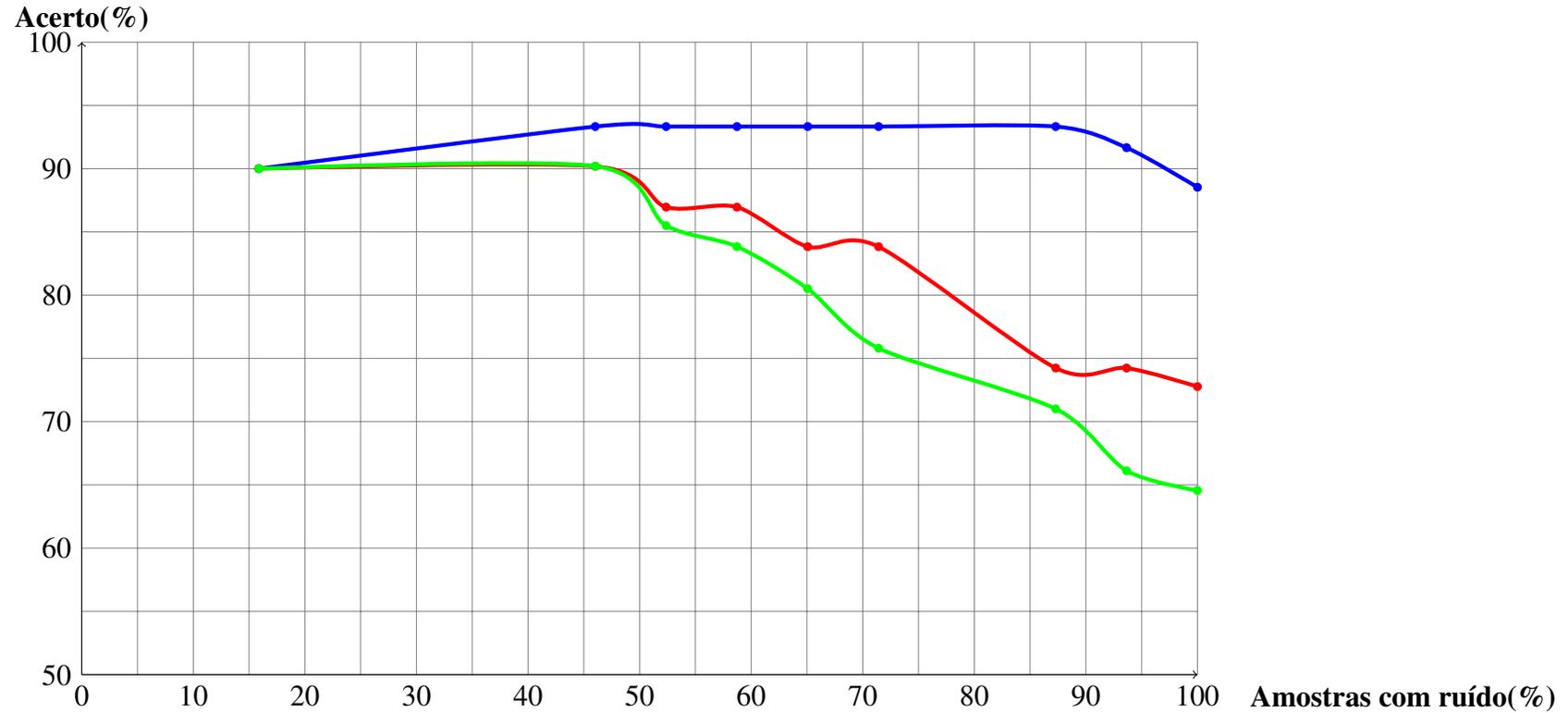


Gráfico 4.3: Inserindo ruído em parte das instâncias (*K-Fold Cross-Validation*)

Legenda do Gráfico 4.3:

Linha 1 (—): Intensidade do ruído = 10%

Linha 2 (—): Intensidade do ruído = 35%

Linha 3 (—): Intensidade do ruído = 65%

A abordagem de *Cross-Validation* influencia a performance do método de maneira significativa, conforme será detalhado na Seção 4.3.4, desta forma, todos os resultados que acabaram de ser vistos (na Tabela 4.8 e no Gráfico 4.3) são o resultado da aplicação da abordagem *K-Fold Cross-Validation*, e, os resultados que se seguem (vistos na Tabela 4.9 e no Gráfico 4.4) serão resultantes da aplicação da abordagem *Leave-One-Out Cross-Validation*.

Tabela 4.9: Etapa 3 - Ruído em parte das instâncias (*Leave-One-Out Cross-Validation*)

Ruído (%)	Amostras Corrompidas (%)	Acerto (%)
10	15,87	100
10	46.03	100
10	52.38	96.83
10	58.73	96.83
10	65.07	96.83
10	71.42	96.83
10	87.30	96.83
10	93.65	96.83
10	100	93.65
35	15,87	100
35	46.03	96.83
35	52.38	85.71
35	58.73	85.71
35	65.07	82.54
35	71.42	82.54
35	87.30	79.37
35	93.65	79.37
35	100	76.19
65	15,87	100
65	46.03	100
65	52.38	88.89
65	58.73	87.30
65	65.07	84.13
65	71.42	79.37
65	87.30	77.78
65	93.65	71.43
65	100	69.84

Os dados da Tabela 4.9 pode ser vistos, de forma mais intuitiva, no Gráfico 4.4.

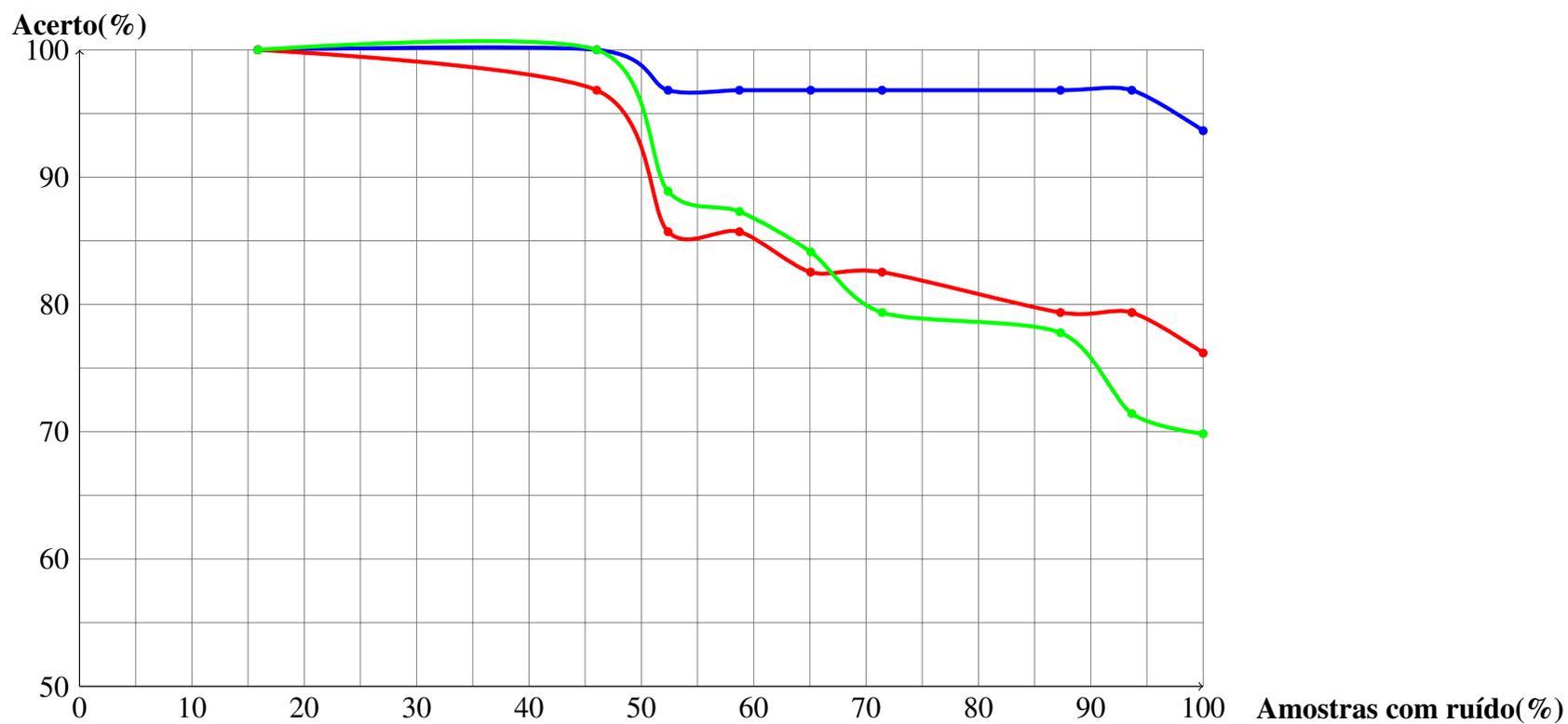


Gráfico 4.4: Inserindo ruído em parte das instâncias (*Leave-One-Out Cross-Validation*)

Legenda do Gráfico 4.3:

Linha 1 (—): Intensidade do ruído = 10%

Linha 2 (—): Intensidade do ruído = 35%

Linha 3 (—): Intensidade do ruído = 65%

Essa bateria de testes foi realizada à partir de três abordagens (a geração de ruído apenas nas instâncias utilizadas para teste, a geração de ruído em todas as instâncias, e a geração de ruído em um percentual das instâncias, que seria utilizado tanto para treinamento quanto para validação). E, a comparação entre essas abordagens mostrou resultados bem diferentes: Quando foi inserido 1% de dados ruidosos apenas nas instâncias utilizadas para teste, isso já foi suficiente para observar que apenas 35% dos testes retornavam valores corretos (conforme pode ser visualizado no Gráfico 4.1). Isso mostra que, nesta abordagem (de gerar ruído apenas nas instâncias utilizadas para teste), os algoritmos utilizados nesta pesquisa se mostraram muito sensíveis ao ruído gerado.

Todavia, considerando a segunda abordagem (de inserir o mesmo nível de ruído em todas as instâncias - as que seriam utilizadas para treinamento, e as que seriam utilizadas para teste), o algoritmo *K-NN* resiste bem ao ruído: mesmo considerando a inserção de 50% de ruído (que implica danificar um a cada dois *pixels* da imagem), foi possível verificar que cerca de 75% dos testes executados ainda retornavam valores corretos. Para constatar essa afirmação observe o Gráfico 4.2.

Por fim, a terceira abordagem (de inserir ruído em um percentual de amostras, que seriam utilizadas, alternadamente, tanto para treinamento quanto para validação) parece ter sido a que apresentou melhor acurácia: observando o Gráfico 4.4 é possível perceber que, quando um ruído com intensidade de 65% foi imposto a 50% das amostras (utilizadas, alternadamente, para treinamento e para validação) o método ainda exibiu cerca de 95% de acerto nos testes.

À partir destas considerações, parece ser possível afirmar que os algoritmos adotados resistem bem ao ruído, desde que ele esteja presente tanto no conjunto de treinamento quanto no conjunto de validação, pois, caso contrário, as distâncias entre as instâncias, no espaço n -dimensional considerado, ficará muito grande, mesmo para baixos percentuais de ruído, como mostra o Gráfico 4.1.

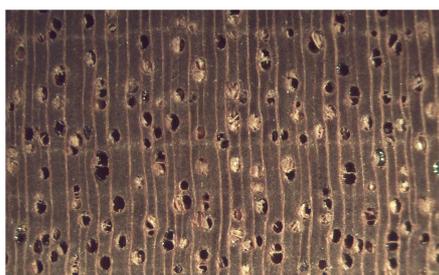
4.2.4 Teste com todas as imagens da Embrapa, usando novas métricas de distância

Após concluir os testes que objetivavam mensurar a resistência do *K-NN* ao ruído (observados na Seção 4.2.3), contando ainda com as mesmas amostras usadas nos testes anteriores (exibidas na Tabela 4.3), e com um “cenário” sem ruído, no qual todos os testes atingiam 100% de acerto, foi implementado um algoritmo que deriva do algoritmo *Kullback Leibler Divergence*, conforme sugestões de Petrou e Sevilla (2006, p. 577), objetivando mensurar, de uma maneira

mais eficiente, as distâncias entre os histogramas.

Em princípio, pode parecer “desnecessário” o esforço de codificar uma nova função de mensuração de distância entre histogramas, em busca de eficiência, posto que, até este momento da pesquisa, com as abordagens de *Cross-Validation*, os testes estavam sempre retornando 100% de acerto. Sobretudo, esse percentual de acertos estava sendo atingido em função de variáveis que deveriam mudar em breve:

- O número de imagens disponíveis, até este momento, ainda era muito pequeno (conforme exposto na Tabela 4.3);
- As imagens disponibilizadas, até este momento, eram muito diferentes entre si, mesmo para uma observação humana, conforme Figura 4.5.



(a) Imagem de Andiroba



(b) Imagem de Goiabão

Figura 4.5: Comparação entre Andiroba e Goiabão (Fonte: Adaptação de imagens fornecidas pela *Embrapa*)

Acreditando que era só uma questão de tempo para dispor de mais imagens, e acreditando que um número bem maior de imagens (contando com espécies que fossem bem parecidas entre si, conforme postulado) seria suficiente para não conseguir mais atingir esse mesmo percentual de acertos (100%), a abordagem de ajustar a função de comparação “deixa de parecer desnecessária”, uma vez que, essa abordagem culminará em resultados com melhor acurácia, quando o banco de imagens passar a corresponder, quantitativamente e qualitativamente, ao que foi postulado.

Segundo Petrou e Sevilla (2006, p. 577), considerando que $f_1(i)$ e $f_2(i)$ representam dois histogramas que precisam ser comparados, então, assumindo que: **A)** i é o i -ésimo binário de cada histograma; **B)** N representa a quantidade de binários de um histograma, que deve ser igual para todos; e, **C)** Todos os binários mencionados possuem a mesma largura, é possível afirmar que, a distância entre dois histogramas pode ser calculada pela Equação 4.1.

$$d(f_1, f_2) \equiv \sum_{i=1}^N f_1(i) \log f_1(i) + \sum_{i=1}^N f_2(i) \log f_2(i) - \sum_{i=1}^N (\log f_1(i) + \log f_2(i)) \log (\log f_1(i) + \log f_2(i)) + 2 \log 2 \quad (4.1)$$

Ainda segundo Petrou e Sevilla (2006, p. 577), para que a aplicação desta Equação exiba resultados satisfatórios é necessário que:

- Os histogramas parametrizados estejam normalizados;
- Os histogramas possuam o mesmo tamanho;
- Os valores das características de cada histograma pertençam a um mesmo domínio de valores.

Na função implementada, com o objetivo de representar a Equação 4.1, foi considerada a seguinte abstração: Se os histogramas não possuírem dimensões iguais, a função retornará -1, caso contrário, retornará a distância entre os histogramas parametrizados, baseada na Equação referenciada (4.1). Esse algoritmo é uma extensão do *Kullback–Leibler Divergence* (equação 2.5 detalhado na Seção 2.1).

Todos os testes de classificação, executados até agora, estavam sendo realizados com a função *knnclassify* do *Matlab*. Esta função, internamente, já possuía vários algoritmos de mensuração de distância entre histogramas, de maneira que, a escolha do algoritmo de mensuração de distâncias só era possível dentro de um leque pré-definido de opções. Neste momento, essa situação gerou o problema descrito na Seção 4.3.1.1, cuja solução adotada foi implementar um *K-NN* próprio, referenciado como *knnufpe*, que permitisse a parametrização da função de mensuração de distâncias entre os histogramas.

A função de mensuração de distâncias, que acabou de ser codificada, conforme especificado por Petrou e Sevilla (2006, p. 577), será referenciada como (*EKLD* - Extensão do *Kullback–Leibler Divergence*). Os testes referentes à performance da nova função podem ser observados nas Tabelas 4.10 e 4.11.

Tabela 4.10: Testando EKLD (*K-Fold Cross-Validation*)

Distância	Normalização	Média (%)	Desvio Padrão	Coef. Variação (%)
Euclidiana	Não	100	0.00	0.00
Euclidiana	Sim	100	0.00	0.00
EKLD	Não	17.86	5.17	29.01
EKLD	Sim	46.43	7.14	15.37

Tabela 4.11: Testando EKLD (*Leave-One-Out Cross-Validation*)

Distância	Normalização	Acerto (%)
Euclidiana	Não	100%
Euclidiana	Sim	100%
EKLD	Não	42,86%
EKLD	Sim	17,86%

Observando os resultados da primeira etapa da quarta bateria de testes, disponíveis nas Tabelas 4.10 e 4.11, foi possível observar que:

- Sempre que a distância euclidiana foi utilizada (independente de trabalhar com o *knnufpe* ou com o *knnclassify* do *Matlab*, e independente das amostras estarem normalizadas ou não) o método retornou 100% de acerto;
- Sempre que a distância *EKLD* (extensão do *Kullback Leibler Divergence*) foi utilizada (independente das amostras estarem normalizadas ou não) o índice de acerto foi baixíssimo. Geralmente inferior a 50%;
- Adicionalmente, trabalhando com o *EKLD*, quando os dados foram normalizados, os índices de acerto ficaram ainda piores.

Pelo que acabou de ser exposto, em tais Tabelas, parece que a função adotada (que implementa o *EKLD* - utilizado para mensurar a distância entre os histogramas) não mostrou uma performance satisfatória. Essa função foi codificada exatamente conforme a Equação descrita no livro de (PETROU; SEVILLA, 2006, p. 577), representada, neste trabalho, pela Equação de número 4.1.

Sobretudo, o livro referenciado informa, explicitamente, que o algoritmo descrito “...is based on the extension of the definition of the *Kullback–Leibler Divergence*...” (PETROU; SEVILLA, 2006, p. 577). Essa informação gerou uma curiosidade: como o *K-NN* se comportaria se fosse parametrizada, como função de comparação entre os histogramas, a função *Kullback-Leibler Divergence* original? À partir desta motivação, a função *Kullback-Leibler Divergence* original foi utilizada como função de comparação, no algoritmo *K-NN* (desenvolvido nesta pesquisa).

O resultado alcançado, parametrizando esta função de mensuração de distância, no referido algoritmo, não poderia ter sido melhor: o método passou a atingir 100% de acerto em todos os

testes, assim como já acontecia quando a função de comparação parametrizada era a distância euclidiana.

A implementação de um K - NN próprio teve uma consequência: foi necessário implementar todas as métricas de distância que precisariam ser utilizadas com ele, pois, essa foi a maneira adotada para resolver o problema citado na Seção 4.3.1.1. Desta forma, todos os códigos, das métricas de distância utilizadas neste trabalho, foram organizados em funções:

- Distância Euclidiana: Algoritmo 2.4;
- Distância EKLD: Equação 4.1;
- Distância KLD: Equação 2.5.

Assim, considerando que mais uma métrica de distância (o KLD) foi incorporada ao trabalho, e que vários testes acerca da tolerância ao ruído já tinham sido executados, então, deste ponto da pesquisa em diante, novos testes só foram realizados quando novas imagens foram disponibilizadas.

4.2.5 Teste mesclando imagens de bancos distintos (Embrapa e UFPR)

Os testes da quinta bateria são os únicos nos quais as imagens da *Embrapa* e da *UFPR* foram mescladas. O objetivo era testar a performance do algoritmo, considerando um banco de imagens heterogêneo (com relação à resolução(tamanho) e ao formato). As imagens da *UFPR* foram disponibilizadas através do banco público de imagens conhecido como *Forest Species Database*.

O algoritmo de classificação considerado durante toda a quinta bateria de testes, foi o K - NN .

As Tabelas A.1 e A.2 descrevem todos os resultados alusivos à quinta bateria de testes. As referidas Tabelas podem ser conferidas no Apêndice A, e o Gráfico 4.5, sumariza os resultados destas Tabelas. As Tabelas vistas logo em seguida (Tabela 4.12 e Tabela 4.13) exibem apenas os melhores resultados das Tabelas vistas no Apêndice em questão.

Para entender as Tabelas 4.12 e 4.13, considere que:

- **GI-GF (Qtd.):** Significa Grupo Inicial - Grupo Final (Quantidade de Imagens)
- **Espécies:** Indica a quantidade de espécies existentes no teste
- **KLD:** É o algoritmo de distância *Kullback Leibler-Divergence* original
- **Euclidiana:** É o algoritmo de distância euclidiana

Essa bateria de testes foi extensa, por isso, as Tabelas que exibem todos os testes, desta bateria, foram dispostas no Apêndice A. Os resultados exibidos, nas Tabelas abaixo, foram selecionados em função de uma acurácia mais evidente, em relação aos demais resultados desta bateria, que podem ser observados no referido apêndice.

Tabela 4.12: Melhores resultados da quinta bateria de testes (K-Fold Cross-Validation)

GI-GF (Qtd.)	Espécies	Distância	Norm.	Média (%)	Desvio Padrão	Coef. Var. (%)
1-9 (63)	9	KLD	Sim	93,33	13.33	14.28
1-10 (83)	10	KLD	Sim	95	10	10.52
1-11 (103)	11	KLD	Sim	96	8	8.33
1-12 (123)	12	Euclidiana	Sim	91,77	6.42	6.99
1-13 (143)	13	Euclidiana	Sim	92,93	6.47	6.97
1-14 (163)	14	Euclidiana	Sim	92,01	3.70	4.02
1-15 (183)	15	Euclidiana	Sim	90,67	4.63	5.11
1-16 (203)	16	KLD	Sim	90,61	4.48	4.95
1-17 (223)	17	KLD	Sim	90,56	4.82	5.33
1-18 (243)	18	KLD	Sim	90,11	4.31	4.79
1-19 (263)	19	Euclidiana	Sim	88,60	3.83	4.33
1-20 (283)	20	KLD	Sim	88,67	5.10	5.75

Tabela 4.13: Melhores resultados da quinta bateria de testes (*Leave-One-Out Cross-Validation*)

GI-GF (Qtd.)	Espécies	Distância	Normalização	Acerto (%)
1-9 (63)	9	KLD	Sim	100
1-10 (83)	10	KLD	Sim	100
1-11 (103)	11	KLD	Sim	100
1-12 (123)	12	Euclidiana	Sim	96,75
1-13 (143)	13	Euclidiana	Sim	97,20
1-14 (163)	14	Euclidiana	Sim	95,09
1-15 (183)	15	Euclidiana	Sim	94,54
1-16 (203)	16	Euclidiana	Sim	95,07
1-17 (223)	17	Euclidiana	Sim	94,62
1-18 (243)	18	Euclidiana	Sim	94,65
1-19 (263)	19	Euclidiana	Sim	94,68
1-20 (283)	20	KLD	Sim	93,64

O Gráfico 4.5 exibe todos os dados desta bateria de testes, e foi construído com os dados das Tabelas A.1 e A.2, que podem ser encontradas no Apêndice A.

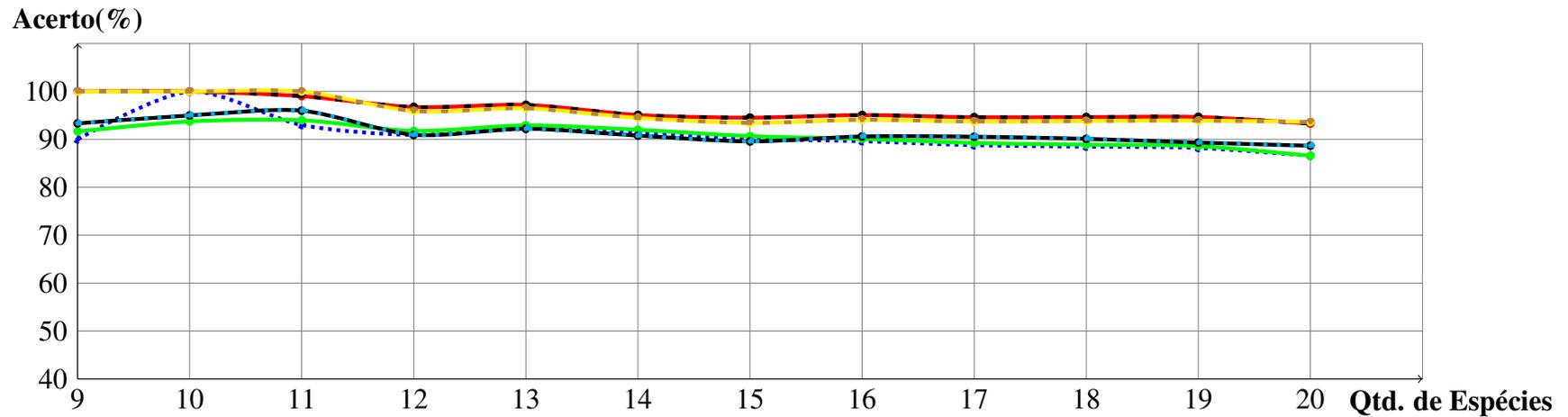


Gráfico 4.5: Teste mesclando imagens da Embrapa e imagens da UFPR

Legenda do Gráfico 4.5:

- Linha 1 (.....): Dist. Euclidiana; 4FCV; Sem normalização.
- Linha 2 (—): Dist. Euclidiana; LOOCV; Sem normalização.
- Linha 3 (—): Dist. Euclidiana; 4FCV; Com normalização.
- Linha 4 (-.-.): Dist. Euclidiana; LOOCV; Com normalização (Sobrepõe completamente a Linha 2).
- Linha 5 (—): Dist. KLD; 4FCV; Sem normalização.
- Linha 6 (—): Dist. KLD; LOOCV; Sem normalização.
- Linha 7 (.....): Dist. KLD; 4FCV; Com normalização (Sobrepõe completamente a Linha 5).
- Linha 8 (-.-.): Dist. KLD; LOOCV; Com normalização (Sobrepõe completamente a linha 6 e parcialmente as Linhas 4 e 2).

O Gráfico 4.5 exibe uma representação dos resultados vistos nas Tabelas do Apêndice A (que possuem todos os dados referentes a esta bateria de testes). Observando esse gráfico, e essas Tabelas, é possível realizar as seguintes inferências:

- É possível perceber que existe uma diferença de performance entre as abordagens de *Cross-Validation*: As abordagens de *Leave-One-Out Cross-Validation* sempre possuem uma acurácia melhor, se comparadas com as abordagens de *K-Fold Cross-Validation*;
- Para esse teste específico, com essa quantidade de instâncias, não existe diferença significativa entre as métricas de distância comparadas, que foram:
 - A distância euclidiana;
 - A distância *KLD*.
- Nesta bateria de testes, a normalização praticamente não fez diferença, pois, no Gráfico 4.5 (cujas linhas representam abordagens diferentes, para as mesmas amostras) das oito linhas disponíveis, existe sobreposição total de informação em seis delas, em situações nas quais a única diferença era o fato de ter, ou não, normalizado as amostras:
 - Linha 2 com linha 4;
 - Linha 5 com linha 7;
 - Linha 6 com linha 8.
- Quando a quantidade de classes envolvidas aumenta, o percentual de acertos tende a baixar, sobretudo, é necessário aumentar significativamente essa quantidade de imagens, para se obter uma mudança impactante no percentual de acertos, pois, no contexto do Gráfico 4.5:
 - Para a abordagem *Leave-One-Out Cross-Validation*: Quando o teste considerou dez classes de imagem, o método estava apresentando 100% de acerto; e, quando essa quantidade de classes foi dobrada (passou para vinte classes), o método ainda apresentava 95% de acerto;
 - Para a abordagem *K-Fold Cross-Validation*: Quando o teste considerou dez classes de imagem, o método apresentava resultados que variavam (dependendo de outras variáveis) de 95% a 100% de acerto; e, quando essa quantidade de classes foi dobrada (passou para vinte classes), o método apresentou resultados (dependendo de outras variáveis) que variavam de 87% a 89% de acerto.

- 112094 (cento e doze mil e noventa e quatro) linhas de código foram necessárias para chegar ao resultado exibido em apenas uma das linhas da Tabela 4.13 (especificamente sua última linha, onde o valor 93,64% é visualizado). Esses dados deixam claro porque o *Leave-One-Out Cross-Validation* não é indicado quando se dispõe de uma base com muitas amostras: O código necessário para se aplicar o *Leave-One-Out Cross-Validation* em muitas amostras fica enorme, e o processamento computacional, que se faz necessário, aumenta em razão diretamente proporcional ao tamanho do código. Por conta dessa situação, essa foi a última bateria de testes na qual o *Leave-One-Out Cross-Validation* foi utilizado;
- O Gráfico 4.5 comprova que as abordagens utilizadas neste trabalho funcionam bem para um banco de imagens heterogêneas (do ponto de vista do tamanho e do formato): O banco utilizado era bem heterogêneo, e foi possível observar que o método apresentou (considerando a abordagem de *Leave-One-Out Cross-Validation*), no ponto pior do Gráfico (quando já estavam sendo consideradas 20 classes diferentes de imagens), cerca de 95% de acerto nos testes.

4.2.6 Teste com todas as imagens da UFPR

Nesta bateria de testes (sexta bateria) foram utilizadas as amostras fornecidas pelo *VRI(UFPR)*. Este centro de pesquisa disponibilizou 2240 imagens, e todas elas foram utilizadas nesta bateria de testes. Em função dessa quantidade de amostras a abordagem *Leave-One-Out Cross-Validation* não foi utilizada, pois, a referida abordagem só é recomendada quando a base de teste não extrapola algumas dúzias de amostras. Em função desta situação, a coluna **Abordagem**, que, nas Tabelas anteriores de resultado, especificava qual a abordagem de *Cross-Validation* utilizada, não será exibida na Tabela 4.14, pois, uma única abordagem de *Cross-Validation* será utilizada.

O algoritmo de classificação utilizado durante toda esta bateria de testes foi o *K-NN* (com $K = 1$), adotando variações nas métricas de distância (distância euclidiana ou distância KLD), e na normalização das amostras.

A Tabela 4.14 descreve todos os resultados alusivos à sexta bateria de testes.

O Gráfico 4.6 sumariza os resultados vistos na Tabela 4.14, mostrando quais abordagens exibiram melhor acurácia durante esta bateria de testes.

Nessa bateria de testes, a única abordagem de Cross-Validation utilizada será o *K-Fold Cross-Validation* (com $K = 4$). E, para entender a Tabela 4.14, considere que:

- **GI-GF (Qtd.):** Significa Grupo Inicial - Grupo Final (Quantidade de Imagens)
- **Espécies:** Indica a quantidade de espécies existentes no teste
- **KLD:** É o algoritmo de distância *Kullback Leibler-Divergence* original
- **Euclidiana:** É o algoritmo de distância euclidiana

Tabela 4.14: Resumo dos resultados dos testes com as imagens da UFPR

GI-GF (Qtd.)	Espécies	Distância	Norm.	Média (%)	Desvio Padrão	Coef. Var. (%)
10-14 (100)	5	Euclidiana	Não	92,00%	7,30	7,93
10-14 (100)	5	Euclidiana	Sim	92,00%	7,30	7,93
10-14 (100)	5	KLD	Não	89,00%	5,03	5,65
10-14 (100)	5	KLD	Sim	89,00%	5,03	5,65
10-19 (200)	10	Euclidiana	Não	87,50%	4,72	5,40
10-19 (200)	10	Euclidiana	Sim	87,50%	4,72	5,40
10-19 (200)	10	KLD	Não	88,00%	3,65	4,14
10-19 (200)	10	KLD	Sim	88,00%	3,65	4,14
10-39 (600)	30	Euclidiana	Não	71,83%	1,91	2,66
10-39 (600)	30	Euclidiana	Sim	72,00%	1,96	2,72
10-39 (600)	30	KLD	Não	74,00%	3,48	4,71
10-39 (600)	30	KLD	Sim	74,00%	3,48	4,71
10-59 (1000)	50	Euclidiana	Não	69,90%	0,66	0,94
10-59 (1000)	50	Euclidiana	Sim	70,00%	0,56	0,80
10-59 (1000)	50	KLD	Não	72,80%	1,03	1,41
10-59 (1000)	50	KLD	Sim	72,80%	1,03	1,41
10-79 (1400)	70	Euclidiana	Não	64,35%	1,96	3,05
10-79 (1400)	70	Euclidiana	Sim	64,57%	1,82	2,82
10-79 (1400)	70	KLD	Não	68,50%	0,94	1,37
10-79 (1400)	70	KLD	Sim	68,50%	0,94	1,37
10-99 (1800)	90	Euclidiana	Não	61,05%	2,56	4,19
10-99 (1800)	90	Euclidiana	Sim	61,16%	1,89	3,10
10-99 (1800)	90	KLD	Não	65,67%	1,72	2,62
10-99 (1800)	90	KLD	Sim	65,67%	1,72	2,62
10-121 (2240)	112	Euclidiana	Não	57,50%	2,44	4,25
10-121 (2240)	112	Euclidiana	Sim	57,63%	1,70	2,95
10-121 (2240)	112	KLD	Não	61,60%	2,43	3,95
10-121 (2240)	112	KLD	Sim	61,60%	2,43	3,95

Os dados da Tabela 4.14 também podem ser visualizados no Gráfico 4.6.

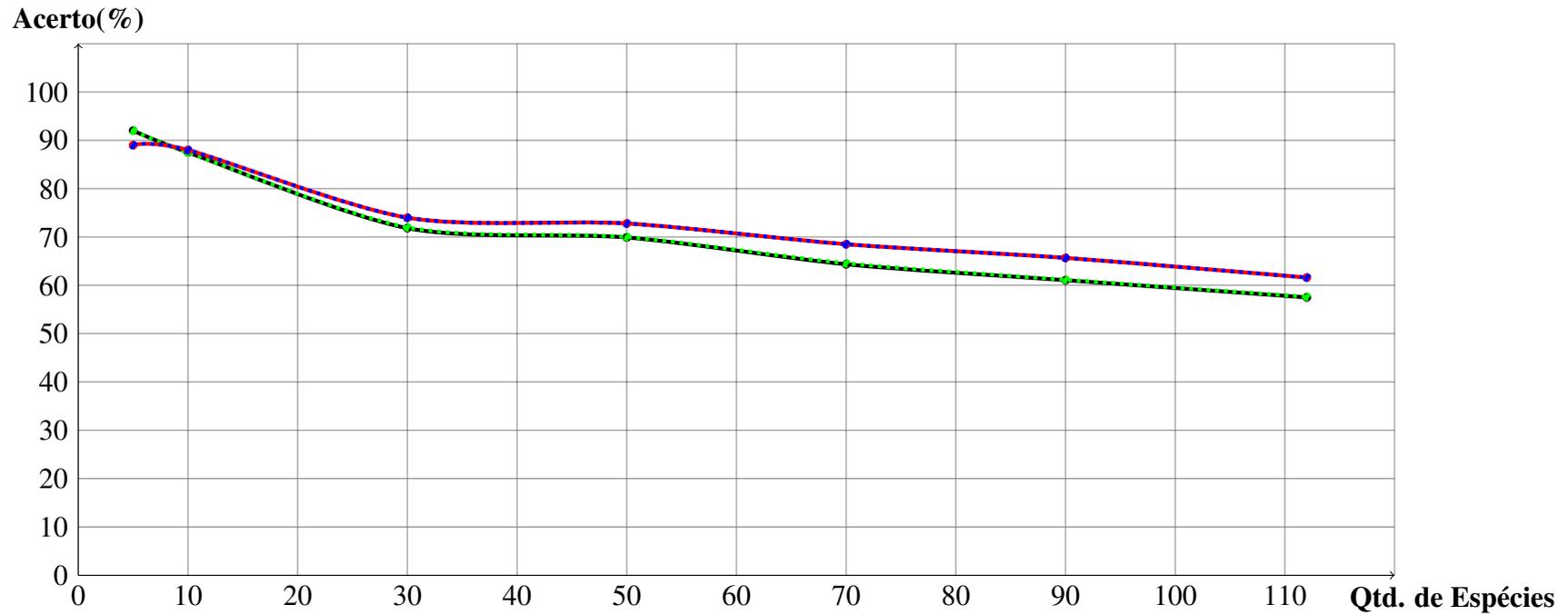


Gráfico 4.6: Teste com todas as 2240 imagens da UFPR

Legenda do Gráfico 4.6:

- Linha 1 (—): Dist. Euclidiana; 4FCV; Sem normalização.
- Linha 2 (.....): Dist. Euclidiana; 4FCV; Com normalização (Sobrepõe completamente a Linha 1).
- Linha 3 (—): Dist. KLD; 4FCV; Sem normalização.
- Linha 4 (.....): Dist. KLD; 4FCV; Com normalização (Sobrepõe completamente a Linha 3).

Conforme mencionado, a Tabela 4.14 mostra os resultados alusivos à esta bateria de testes (sexta bateria de testes), e o Gráfico 4.6 exibe os mesmos resultados, de uma forma mais intuitiva. Assim, com base nesta tabela, e neste gráfico, parece possível fazer as seguintes inferências:

- Esta bateria de testes (a sexta) utilizou 2240 imagens, uma quantidade bem maior que a utilizada na quinta bateria de testes (283 imagens), e isso foi suficiente para perceber alguns detalhes que não puderam ser percebidos na quinta bateria de testes, conforme abordado em observações seguintes;
- Todas as imagens consideradas, neste teste, possuíam exatamente o mesmo tamanho e o mesmo formato, então, de maneira contrária à quinta bateria de testes, na sexta é impossível falar de banco de dados heterogêneo: ele é completamente homogêneo;
- Nesta bateria de testes (a sexta), contando com uma quantidade expressiva de imagens, ficou bem clara a diferença de performance entre as métricas de distância: A distância *KLD* apresentou uma acurácia melhor em quase todos os momentos, quando comparada com a distância euclidiana; Provavelmente, essa diferença de performance não foi percebida na quinta bateria de testes porque ela contou com uma quantidade de imagens bem menor. Para perceber essa diferença de performance, observe, no Gráfico 4.6, o seguinte:
 - Linha 1 (distância euclidiana): Resultado inferior na maior parte do gráfico;
 - Linha 2 (distância euclidiana): Resultado inferior na maior parte do gráfico;
 - Linha 3 (distância *KLD*): Resultado superior na maior parte do gráfico;
 - Linha 4 (distância *KLD*): Resultado superior na maior parte do gráfico.
- É possível afirmar que, nesta bateria de testes (sexta bateria), a normalização não fez diferença alguma. Perceba, observando o Gráfico 4.6, que quando as outras variáveis eram mantidas, e apenas a normalização era acrescentada, ou removida, as linhas resultantes ficavam sobrepostas:
 - Linha 1 e linha 2;
 - Linha 3 e linha 4.
- O método, proposto neste trabalho, se mostrou válido para interagir com imagens homogêneas;

- Como já se esperava, à medida que a quantidade de espécies aumenta, o número de acertos diminui. Desta forma, se este trabalho for estendido, sem novos ajustes no método proposto, para criação de uma ferramenta de apoio para a fiscalização nas alfândegas, é interessante que esta ferramenta não trabalhe com mais de dez classes de imagens, pois, sem extrapolar essa quantidade de classes, o método alcança cerca de 90% de acerto. Para constatar esses percentuais de acerto, observe o Gráfico 4.6. Se o percentual de acertos, da ferramenta em questão, começar a baixar muito, em função do aumento na quantidade de classes, ela pode não representar uma ferramenta viável para os agentes alfandegários da amazônia;
- Por fim, uma solução válida, considerando que existem muito mais que dez classes de imagem a serem classificadas na amazônia, pode ser observada na Seção 6.5, que representa uma das sugestões para trabalhos futuros;

4.3 Análise

Esta seção exibe inferências de âmbito geral, que foram desenvolvidas observando todas as baterias de teste.

A análise é o processo mais importante do trabalho: uma interpretação equivocada pode invalidar as conclusões de uma pesquisa, gerando inferências absurdas, como a que pode ser percebida na metáfora do pesquisador sem talento, que *”deduziu que uma aranha sem patas é surda, apenas porque ela não conseguia mais obedecer ao comando **ande**, depois que todas as suas patas foram arrancadas.”*

Neste trabalho, a análise foi feita através das seguintes abordagens:

- Observando as características das amostras (especificamente tamanho, formato e presença de ruído) selecionadas para cada teste, de cada bateria de testes: Amostras com características diferentes produzem resultados diferentes, motivo pelo qual, as características, que determinam a qualidade das amostras, representam outras variáveis deste processo, e são sensíveis ao ruído, conforme se constata observando os resultados da terceira bateria de testes (disponíveis na Seção 4.2.3);
- Observando as parametrizações adotadas para cada algoritmo utilizado, em cada teste, de

cada bateria de testes: todos os algoritmos utilizados, neste trabalho, estão encapsulados¹ em funções. A alteração de um único parâmetro, na chamada destas funções, pode alterar completamente o resultado final do processo.

- Anotando as correlações percebidas entre as alterações de cada variável e a performance do método: esse é um dos papéis mais importantes da análise, pois, indica de que forma o método deve ser reajustado para apresentar uma acurácia melhor;
- Observando, e anotando, as respostas do método, para cada teste, de cada bateria de testes: essas anotações representam insumos para a análise;
- Observando os Gráficos resultantes dos teste: eles facilitam o entendimento dos resultados alcançados;

4.3.1 Dificuldades encontradas

O objetivo desta seção é abordar as dificuldades encontradas, e descrever, de maneira objetiva, as alternativas, adotadas ou sugeridas, para resolver cada problema citado. As subseções, desta seção, que tratam de abordagens que já foram experimentadas para solucionar determinada dificuldade relatada, possuirão o nome **Solução adotada**, e, analogamente, as subseções, desta seção, que tratam de abordagens que apenas estão sendo sugeridas, e ainda não foram experimentadas para solucionar determinada dificuldade relatada, possuirão o nome **Solução sugerida**.

4.3.1.1 *K-NN* do *Matlab* (*knnclassify*) não mensurava distâncias pela Equação 4.1

Descrição

Petrou e Sevilla (2006, p. 577) recomendam, para mensurar a distâncias entre dois histogramas, uma função de mensuração de distância que deriva da função conhecida como *Kullback Leibler Divergence*, vista na Equação 4.1.

Quando ficou decidido que essa Equação (4.1) seria utilizada nesse trabalho, um impasse foi encontrado: a função *knnclassify* do *Matlab* (que utiliza, internamente, o algoritmo *K-NN*)

¹Segundo Deitel e Deitel (2005, p. 58), Encapsular significa fornecer uma *interface* para executar uma tarefa que já está programada

não permitia, nas parametrizações possíveis, a comparação de histogramas através da referida equação.

Solução adotada

A postura adotada foi a de implementar um K - NN próprio, que permitisse a parametrização da função de comparação entre os histogramas. Desta forma, a função de comparação seria construída fora da função K - NN , e referenciada como um parâmetro, assim, seria possível utilizar qualquer função de comparação que se julgasse necessária durante a pesquisa. E isso foi feito: a assinatura da nova função K - NN , em princípio, foi construída imitando a assinatura do método *knnclassify* do *Matlab*. Pelas necessidades mencionadas, dois novos parâmetros foram acrescentados, objetivando disponibilizar as funcionalidades necessárias:

- Um parâmetro para informar a função de comparação a ser utilizada;
- Um parâmetro para informar o K (número de vizinhos mais próximos a serem considerados).

Em suma, o objetivo principal da implementação de um K - NN próprio foi a possibilidade de parametrizar a função de comparação entre histogramas, pois, o K já podia ser parametrizado na função disponibilizada pelo *Matlab* (*knnclassify*). Vale ressaltar que o K - NN implementado neste trabalho segue, à risca, o algoritmo de Mitchell (1997, p. 232), referenciado nesta dissertação como algoritmo 2.1.

4.3.1.2 Qualidade das imagens utilizadas

Descrição

Conforme abordado na Seção 4.1.1 (que trata das ferramentas da pesquisa), todas as imagens fornecidas pela *Embrapa* foram coletadas diretamente da xiloteca que existe na *Embrapa*. Adicionalmente, todas as amostras de madeira, selecionadas para captura das imagens, passaram por procedimentos padronizados de tratamento e armazenamento, conforme organização respeitada pela xiloteca em questão, desta forma, todas as superfícies fotografadas, e disponibilizadas para esse trabalho, possuem padrões semelhantes de qualidade (Fonte: *Embrapa*).

A qualidade das imagens do banco público *Forest Species Database* foi alcançada através

de um microscópio de modelo *Olympus Cx40*, considerando um *zoom* de 100 vezes (Fonte: <http://web.inf.ufpr.br/vri/image-and-videos-databases/forest-species-database>).

Nenhuma das informações exibidas nesta seção representará uma novidade para o leitor, caso ele já tenha lido a seção que trata das ferramentas da pesquisa (seção 4.1.1), sobretudo, o objetivo nesta seção é fazer o leitor recordar o que já foi mencionado, para entender o "problema": todas as imagens utilizadas para treinamento (tanto as que foram fornecidas pela *Embrapa* quanto as que foram fornecidas pelo *VRI(UFPR)*, através do banco público *Forest Species Database*), seguem padrões de qualidade que podem não ser alcançados pelas imagens que serão capturadas em uma alfândega, durante uma fiscalização real. Para entender essa assertiva, sem precisar refletir muito, lembre que as imagens fornecidas pela *Embrapa* foram preparadas através de um equipamento conhecido como micrótomo (visto na Figura 3.2a), inexistente nas alfândegas. Já no caso do banco público de imagens, fornecido pela *UFPR*, é possível elencar preocupações semelhantes, entendendo que, se os agentes alfandegários não dispuserem de aparelhos com qualidade semelhante aos que foram utilizados para montar o banco público de imagens, as chances de erro aumentam.

Objetivando estabelecer uma analogia para facilitar o entendimento desta situação, imagine que um sistema foi criado para classificar carros de acordo com suas imagens, adicionalmente, considere que todo o banco de imagens de treinamento foi montado com imagens capturadas ao meio dia, contando com o auxílio da luz solar (que majora a qualidade das imagens). Considere que o sistema hipotético, após milhares de testes, está atingindo 95% de acertos. O questionamento que se deseja estabelecer, neste momento, é o seguinte: *será que se o mesmo sistema for testado na presença de chuva, ou no período noturno, um percentual de acertos semelhante será alcançado?* Os pingos de chuva, ou a diferença radical de luminosidade, podem atuar como ruído. Analogamente, não parece interessante utilizar imagens de alta qualidade (como as que foram utilizadas nessa pesquisa) para treinar um classificador que deverá ser utilizado para classificar imagens de qualidade aleatória.

Solução sugerida

Em princípio, para a xiloteca, quanto maior a qualidade das imagens melhor, entretanto, para os problemas de visão computacional, é importante que exista similaridade entre as imagens que formam o conjunto de treinamento, e as imagens que formam o conjunto de validação. A "solução" sugerida, nesse trabalho, é organizar um banco de imagens, para treinamento do classificador, montado com imagens capturadas nas próprias alfândegas, pois, esta abordagem

deve majorar a similaridade entre o conjunto de imagens utilizadas para treinamento e para classificação.

4.3.2 Limitações e análise de validade

4.3.2.1 Das limitações

É possível perceber, nos gráficos disponíveis neste trabalho (notadamente nos gráficos 4.5 e 4.6), que quando o número de classes (categorias) de imagens, utilizadas para treinamento, aumenta, o percentual de acertos diminui, mesmo a despeito dos valores atribuídos às outras variáveis (Abordagem de *Cross-Validation*, Normalização, etc). E, não é difícil entender o motivo: imagine que uma moeda, não viciada², seja jogada para cima; a chance de acertar que lado da moeda ficará visível, quando ela ficar parada no chão, é de $1/2 = 50\%$. Analogamente, imagine que um dado, não viciado, foi jogado para cima, a chance de acertar que lado será exibido, quando ele ficar parado no chão, é de $1/6 = 16,66\%$. Obviamente, esses percentuais, para ambas as analogias, são calculados considerando um palpite aleatório, analogamente, para um método que trabalha com aprendizado baseado em instância (detalhado na Seção 2.1.8), que, pela própria concepção, faz uso de modelos matemáticos para categorizar o que lhe é apresentado, o aumento na quantidade de possibilidades de resposta pode implicar um decréscimo no percentual total de acertos, pelas mesmas razões lógicas das analogias estabelecidas: considerando que exista apenas uma resposta certa, quanto maior o leque de respostas possíveis, menor a possibilidade de acertar a resposta correta, à semelhança de um jogo de azar, quando o número de respostas possíveis aumenta.

Observando o gráfico 4.6, resultante da sexta bateria de testes, na qual chegaram a ser consideradas até 112 classes diferentes de imagem, foi possível perceber que quando a quantidade de classes de imagem foi menor ou igual a 10, os índices de acerto estavam em torno de 90%, entretanto, quando foram utilizadas 100 ou mais espécies, os índices de acerto estavam próximos de 60%.

Neste contexto, a metodologia desenvolvida neste trabalho parece adequada para se trabalhar com até dez classes diferentes de imagem. Sobretudo, não existem subsídios para informar, de maneira determinística, o que é adequado ou não: isso será uma função da tolerância de quem “comprar a idéia” de utilizar o método proposto.

²A expressão “não viciado(a)” é comum em livros de estatística, como o de Freund e Simon (2000), e significa que nenhum recurso adicional foi instaurado para alterar a aleatoriedade natural do evento em análise

4.3.2.2 Da validade dos métodos utilizados

Nesta trabalho, foi possível dispor de bases de dados com características bem diferentes (no que diz respeito a tamanho, e formato das imagens), e, a metodologia adotada produziu resultados semelhantes em todas as bases de dados. Adicionalmente, na quinta bateria de testes, foram mescladas imagens da *Embrapa* e da *UFPR*, e, mesmo à despeito de se considerar conjuntos de imagens com formatos distintos, o método continuou exibindo percentuais de acerto semelhantes, no momento da categorização.

Desta forma, o método é válido para bases de imagens com formatos completamente distintos, conforme experimentado na quinta bateria de testes, e, também é válido para bases de imagens homogêneas (com mesmo tamanho e formato), conforme experimentado na sexta bateria de testes.

Sobretudo, o universo de bases de dados utilizado neste trabalho (apenas duas bases: uma fornecida pela *Embrapa* e outra pelo *VRI(UFPR)*) é pequeno demais para fazer afirmações determinísticas acerca da validade deste método em outros contextos.

4.3.3 Sobre a normalização

Parece possível afirmar, pela análise dos resultados, que a normalização (explanada na Seção 2.1.7) foi o elemento menos influente neste trabalho.

Observando o Gráfico 4.5 (referente à quinta bateria de testes), especificamente as linhas 1 e 3 (que representam testes similares, cuja única diferença é a normalização, presente na linha 1 e ausente na linha 3) é possível perceber que a linha 1 começou desenvolvendo uma performance melhor, entretanto, logo em seguida passou a exibir uma performance ligeiramente inferior, e se manteve assim durante o resto da evolução do gráfico. O ponto capital nesta observação é perceber que não existiu uma diferença significativa de performance causada pela presença, ou pela ausência, da normalização. É possível chegar à mesma conclusão fazendo as seguintes comparações:

- Linha 2 com linha 4: Se sobrepõem completamente;
- Linha 5 com linha 7: Se sobrepõem completamente;
- Linha 6 com linha 8: Se sobrepõem completamente.

A primeira comparação realizada (linha 1 com linha 3) é a única na qual os resultados não são exatamente iguais. Observando, agora, o Gráfico 4.6 (referente à sexta bateria de testes) faça as seguintes comparações:

- Linha 1 com linha 2: Se sobrepõem completamente;
- Linha 3 com linha 4: Se sobrepõem completamente.

Nesse cenário, parece possível inferir que a normalização não foi um fator impactante para o performance dos resultados alcançados.

4.3.4 Sobre as abordagens de *Cross-Validation*

Foi possível deduzir, considerando um mesmo conjunto de amostras, que a alteração da abordagem de *Cross-Validation* é suficiente para alterar o performance dos resultados: quando o *Leave-One-Out Cross-Validation* é adotado, os resultados apresentam uma acurácia melhor em relação à utilização do *K-Fold Cross-Validation*, ainda que sejam consideradas as mesmas amostras em ambas as abordagens.

Existe uma lógica por trás desses resultados: sempre que a abordagem *Leave-One-Out Cross-Validation* é adotada, o espaço amostral (número de amostras) do conjunto de treinamento fica maior, em relação à abordagem *K-Fold Cross-Validation*, e, um método “mais treinado” deve responder melhor. Essa assertiva se constata pela observação dos seguintes gráficos:

- 4.2: Especificamente pela observação da linha 2;
- 4.5: Especificamente pela observação das linhas 2, 4, 6 e 8.

Essa percepção é suficiente para estabelecer um raciocínio importante: quando um classificador for disponibilizado, para resolver problemas práticos em uma situação real (como servir de ferramenta de apoio para a fiscalização nas alfândegas brasileiras), é importante deixar a base utilizada para treinamento tão numerosa quanto possível, objetivando majorar a eficiência do referido classificador.

4.3.5 Sobre as métricas de distância

Conforme abordado na Seção 4.3.1.1, Petrou e Sevilla (2006, p. 577) recomendam, para mensurar a distâncias entre dois histogramas, uma função de mensuração de distância que deriva da função conhecida como *Kullback Leibler Divergence*. A função em questão (recomendada por estes autores) esta representada pela Equação 4.1, encontrada na Seção 4.2.4, e foi referenciada, neste tabalho, como distância *EKLD* (Extensão do *Kullback Leibler Divergence*).

A implementação de distância *EKLD* não surtiu resultados satisfatórios neste trabalho: sempre que a distância *EKLD* foi utilizada (independente das amostras estarem normalizadas ou não) o índice de acerto foi baixíssimo. Geralmente inferior a 50%. Esse algoritmo foi codificado conforme descrição encontrada no livro de (PETROU; SEVILLA, 2006, p. 577).

Assumindo que a distância *EKLD* (recomendada pelos autores Petrou e Sevilla (2006)) não se mostrou adequada para os objetivos deste trabalho, a postura adotada foi a de utilizar a função que deu origem à função recomendada: o *KLD* original. Adicionalmente, conforme exposto, em várias seções deste trabalho, a distância euclidiana (explanada pelo algoritmo 2.4 na Seção 2.1) também foi utilizada em parte dos testes desenvolvidos.

Observando as estatísticas da quinta bateria de testes (que mistura as imagens fornecidas pela *Embrapa* com as imagens fornecidas pelo *VRI(UFPR)*, vista na Seção 4.2.5), em várias situações distintas, mantendo todas as demais variáveis (*Abordagem de Cross-Validation*, *Normalização*, etc), e, alterando apenas a métrica de distância, principalmente sob o prisma dos gráficos, não houve diferença significativa de performance, quando a distância euclidiana foi substituída pela distância *KLD*. Para chegar a essa mesma conclusão, faça as seguintes comparações, observando o Gráfico 4.5:

- A linha 5 com a linha 1;
- A linha 3 com a linha 7;
- A linha 2 com a linha 6;
- A linha 4 com a linha 8.

Perceba que, em todos esses casos, as linhas citadas não se sobrepõem na maior parte do tempo, entretanto, sempre estão muito próximas; E, em todos esses casos, a primeira linha citada apresenta um performance ligeiramente melhor, motivo pelo qual, horas a distância *KLD*

apresentou uma acurácia melhor, horas a distância euclidiana apresentou uma acurácia melhor. Por este contexto, para essas amostras, sob o prisma dos resultados alcançados, foi possível afirmar que as métricas de distância analisadas, especificamente para a quinta bateria de testes, foram equivalentes.

Observando as estatísticas da sexta bateria de testes (que utiliza apenas as imagens fornecidas pelo *VRI(UFPR)*), em várias situações distintas, mantendo todas as demais variáveis, e alterando apenas a métrica de distância, principalmente sob o prisma dos gráficos, foi percebida uma diferença significativa de performance, quando a distância euclidiana foi substituída pela distância *KLD*. Para chegar a essa mesma conclusão, faça as seguintes comparações, observando o Gráfico 4.6:

- A linha 3 com a linha 1;
- A linha 4 com a linha 2.

Pela análise sugerida, o *KLD* se mostrou mais eficiente que a distância euclidiana. Provavelmente, essa diferença de performance não foi tão visível na quinta bateria de testes porque só estavam sendo utilizadas 283 imagens. Já a sexta bateria de testes utilizou todas as imagens fornecidas pelo *VRI(UFPR)* (2240 imagens), e a diferença de performance, quando a métrica de distâncias foi alternada, ficou mais evidente.

Sobretudo, foi bem perceptível, principalmente quando se trabalhava com mais de 2000 amostras, que o computador encerrava a bateria de testes cerca de 10 vezes mais rápido utilizando o *KLD* como métrica de distância: em um dos últimos testes realizados, com as 2240 imagens, enquanto o teste que utilizava a distância *KLD* como métrica demorou aproximadamente 30 segundos para ser concluído, o teste que utilizava a distância euclidiana demorou aproximadamente 5 minutos para ser concluído. Isso mostrou que a distância *KLD* exige muito menos do processador que a distância euclidiana.

Considerando que os resultados da classificação foram melhores quando o *KLD* foi utilizado (conforme constatado na sexta bateria de testes), e considerando que ele exige menos do processador, caso não sejam estudadas outras métricas de distância, conforme sugerido na Seção 6.1, o *KLD* seria a métrica sugerida para trabalhos futuros.

CAPÍTULO 5

CONCLUSÕES

Para o método desenvolvido neste trabalho, algumas variáveis foram mais importantes que outras, como pode ser percebido nas subseções da seção 4.3.

O ruído pode ter um impacto muito significativo nos resultados da classificação, conforme explanado na Seção 4.2.3.

Considerando que este trabalho seja estendido, sem nenhuma melhoria no classificador sugerido, entendendo que a performance do classificador também é uma função da quantidade de classes de resposta que o classificador “conhece”, então, conforme sugerido na Seção 4.3.2.1, é interessante não trabalhar com mais que dez classes, e, se necessário, utilizar mais de um classificador, de acordo com a recomendação vista na Seção 6.5.

O classificador construído neste trabalho se mostrou igualmente eficiente para bases com imagens homogêneas (nas quais as imagens possuem as mesmas dimensões e formatos) e para as bases com imagens heterogêneas (nas quais as imagens não possuem as mesmas dimensões e formatos).

A normalização foi o elemento que menos influenciou a performance do classificador, conforme detalhado na Seção 4.3.3.

As técnicas de *Cross-Validation* influenciam bastante nos resultados, como pode ser percebido lendo a seção 4.3.4.

As métricas de distância influenciam duas coisas, muito importantes, no processo de categorização:

- O índice de acertos alcançado pelo método, como pode ser percebido pela observação da acurácia, nos gráficos:
 - Gráfico 4.2: Especificamente pela observação da linha 2;
 - Gráfico 4.5: Especificamente pela observação das linhas 2, 4, 6 e 8;
 - Gráfico 4.6: Especificamente pela observação das linhas 3 e 4.

- As exigências de processamento computacional: Conforme explanado na Seção 4.3.5, a métrica de distância referenciada como distância *KLD* exige, aproximadamente, dez vezes menos, do processador, que a métrica de distância referenciada como distância euclidiana.

O aumento na quantidade de amostras, disponíveis para um teste específico, é suficiente para tornar mais evidentes as diferenças de performance entre as abordagens utilizadas em cada teste: Na quinta bateria de testes, não foi percebida uma diferença significativa, nos percentuais de acerto do classificador, quando a métrica **distância euclidiana** foi substituída pela métrica **distância *KLD***; Já na sexta bateria de testes, a métrica de **distância *KLD*** se mostrou, em quase todos os momentos, com uma acurácia bem melhor que a métrica **distância euclidiana**. É importante lembrar que a quinta bateria de testes contou com, apenas, 283 imagens, enquanto a sexta bateria de testes contou com 2240 imagens. Ficou bem notório, neste trabalho, que quando o número de amostras aumenta os testes revelam mais informações.

Um classificador baseado em instância pode perceber coisas à semelhança de um olho humano: Na primeira bateria de testes, o método só confundia as imagens de Cedro com as imagens de Faveira, e vice versa; Observando todas as imagens disponíveis, sob a ótica do olho humano, a única imagem que se parecia um pouco com o Cedro era justamente a Faveira, e ambas eram muito diferentes de todas as demais; Assim, foi possível perceber que a “confusão”, que foi feita pelo método, também seria a única possível sob o prisma do olho humano.

Um classificador baseado em instância também pode perceber coisas que já não seriam tão óbvias para um olho humano: Na terceira bateria de testes, na etapa que inseria ruído tanto nas amostras de treinamento quanto nas amostras de validação, quando já estava sendo considerado 50% de degradação (que significa que metade dos *pixels*, de cada imagem, estavam estragados) o método ainda conseguia exibir cerca de 75% de acerto. E, sob o prisma do olho humano, num processo desprovido de qualquer aparato tecnológico, considerando essas mesmas imagens, com esse mesmo nível de degradação, seria muito difícil atingir 75% de acerto na categorização, à semelhança da performance demonstrada pelo método. K-Fold Cross-Validation

Pela sensibilidade que este método de classificação demonstrou, em relação ao ruído, notadamente durante a primeira etapa da terceira bateria de testes (que pode ser vista na Seção 4.2.3, ou no gráfico 4.1), foi possível intuir que as imagens disponibilizadas pela *Embrapa* e pelo *VRI(UFPR)* não estavam adequadas para construir um classificador que precisasse atuar em situações reais (na fiscalização, em uma alfândega): Conforme explanado na Seção 4.3.1.2,

não existem máquinas, nas alfândegas brasileiras, para capturar imagens com a mesma qualidade das que foram utilizadas nesse estudo, e, essa diferença notória de qualidade (nas imagens que seriam submetidas ao método) atuaria como um forte ruído, que, possivelmente, inviabilizaria a utilização deste classificador. A seção 4.3.1.2 exibe uma sugestão que pode resolver esta situação.

Existem dezenas de formas de se obter resultados similares, aos que foram alcançados na construção deste método, através de abordagens diferentes. É possível constatar a veracidade desta assertiva pela análise dos trabalhos relacionados, disponíveis na Seção 2.3.

Foi possível perceber, analisando a seção de trabalhos relacionados (seção 2.3), que o índice de trabalhos similares é muito maior nos países que possuem o mesmo problema do Brasil: tráfico ilegal de madeira.

Existe uma necessidade real, e urgente, acerca da extensão deste protótipo (desenvolvido com o *Matlab*), pois, a criação de uma ferramenta concebida para coibir o tráfico ilegal de madeiras seria importante, notadamente para os agentes alfandegários, conforme necessidades detalhadas na Seção 1.1.1.

Considerando a extensão deste trabalho, para construção de um categorizador eficiente para cenários reais, o estudo que foi realizado é apenas o início: é necessário fazer uma série de testes adicionais, para se chegar a um categorizador com melhor acurácia.

As madeiras mais procuradas, pelos atores do tráfico ilegal, estão compreendidas entre as nove que foram disponibilizadas pela *Embrapa*.

A análise dos resultados deste trabalho, disponíveis, principalmente, na Seção de análise (seção 4.3), detalha a performance do método construído, em diversos cenários, e serve de insumo para inferir que esta abordagem é válida para a classificação almejada, considerando as limitações descritas na Seção 4.3.2.1. Estes resultados serão utilizados como subsídios para projetos futuros.

O método, cujos resultados estão em análise, representa a construção apontada como objetivo precípua deste trabalho, e servirá de norte para a classificação automatizada de madeiras em cenários reais, considerando as limitações identificadas na Seção 4.3.1.2.

TRABALHOS FUTUROS

O objetivo, nesta seção, é direcionar trabalhos correlatos ou a extensão deste mesmo trabalho.

6.1 Estudar outras métricas de distância

Conforme visto na Seção 4.3.5, a métrica de distâncias que mostrou os melhores resultados, tanto do ponto de vista do percentual de acertos (na classificação retornada pelo *knnufpe*), quanto do ponto de vista do processamento computacional, foi a métrica *KLD*.

Sobretudo, existem muitas métricas de distância, além da distância euclidiana e da distância *KLD* (testadas neste trabalho), tais como as que foram citadas por (DUDA; HART; STORK, 2001, p. 188):

- *Métrica de Minkowski;*
- *Distância de Manhattan;*
- *Métrica de Tanimoto.*

Entendendo que outras métricas também podem ser utilizadas para mensurar a distância entre dois elementos, em um espaço n -dimensional R^n , parece sensato sugerir a realização de novas baterias de teste, adotando outras métricas de distância.

6.2 Realizar novos testes com outro descritor de texturas

Os autores Vieira et al. (2012, p. 1-5), conforme relatos encontrados na Seção 2.1.2, realizarem testes com os descritores de textura:

- *LBP - Local Binary Pattern*;
- *GLCM - Gray Level Cooccurrence Matrices*;
- *SDH - Sum and Difference Histograms*;
- *FLBP - Fuzzy Local Binary Pattern*;
- *SIFT - Scale Invariant Feature Transform*;
- *LFP - Local Fuzzy Pattern*.

E informaram que o método *LFP - (Local Fuzzy Pattern)* tem maior poder discriminativo que o *LBP*, então, com base na experiência relatada no trabalho destes autores, parece conveniente sugerir novos testes utilizando o *LFP* como descritor de textura.

6.3 Ajustar a qualidade das imagens em função das situações reais

Conforme explanado na Seção 4.3.1.2, para uma xiloteca, quanto maior a qualidade das imagens melhor, entretanto, para sistemas que trabalham com *Machine Vision*, é importante que exista similaridade entre as imagens que formam o conjunto de treinamento, e as imagens que formam o conjunto de validação. E, como consequência de uma série de fatores, não houve tempo hábil para montar uma base de treinamento mais adequada. A recomendação é organizar um banco de imagens, para treinamento do classificador, montado com imagens capturadas nas próprias alfândegas, pois, esta abordagem deve majorar a similaridade entre o conjunto de imagens utilizadas para treinamento e para classificação.

E, lembrando, conforme constatado na terceira bateria de testes (seção 4.2.3), a abordagem utilizada neste trabalho é muito sensível ao ruído, quando ele está presente apenas no conjunto de validação, e não está presente no conjunto de treinamento. Considerando que as imagens do mundo real (coletadas nas alfândegas na hora da fiscalização) terão uma qualidade muito inferior às que foram fornecidas pela *xiloteca* da *Embrapa*, a expectativa é que essa diferença de qualidade represente um forte ruído, desencadeando uma péssima performance, que justifica esta sugestão.

6.4 Implementação de uma versão para *Smart Phone*

Em princípio, o objetivo era sugerir que este trabalho fosse estendido com a disponibilização de um *Web Service* que oferecesse o serviço de categorização das imagens, e fosse utilizado por um sistema, instalado em um computador com acesso à *Internet*, ou por qualquer dispositivo móvel (celular, *tablet*, entre outros) que se conectasse à *Internet*. Na prática, de acordo com o cenário detalhado pela *Embrapa*, a identificação das madeiras precisa ser feita nas alfândegas brasileiras. Neste cenário (nas alfândegas brasileiras), diversas vezes não será possível contar com o acesso à *Internet*, pois, algumas alfândegas se localizam em locais nos quais não existe infra estrutura para transmissão do sinal (Fonte: *Embrapa*). Desta maneira, parece conveniente sugerir que a aplicação utilizada, para categorizar as madeiras, seja desenvolvida para um *Smart Phone*, que aloque, em um banco de dados próprio, todas as instâncias de treinamento consideradas, pois, desta forma, “todo o conhecimento”, necessário para a identificação de um grupo específico de espécies, já estaria nas mãos do agente, e não seria necessário contar com um sinal de *Internet*.

6.5 Criar vários classificadores

Conforme explanado na Seção 1.1.1, atualmente, o tráfico ilegal de madeira na Amazônia acontece, quase sempre, da seguinte maneira: Os fraudadores criam “notas frias”, ou seja, notas que especificam madeiras cuja extração é lícita, e que não correspondem, na realidade, as madeiras que estão sendo transportadas por eles. E, os agentes alfandegários, por ignorância acerca da classificação da carga, são obrigados a liberar a passagem (Fonte: *Embrapa*).

Os colegas da *Embrapa* também informaram que as madeiras identificadas nas “notas frias” se parecem muito com as que estão, de fato, sendo transportadas. Essa postura, de declarar uma carga parecida com a que está sendo transportada, é adotada pelos fraudadores com o objetivo de dificultar a fiscalização, e esse objetivo tem sido atingido.

Adicionalmente, ainda segundo informações da *Embrapa*, embora a Amazônia disponha de milhares de espécies de árvores, a maioria delas não representam objeto de interesse para o tráfico ilegal, que se concentra em um número pequeno de espécies.

Com base nessas informações, a sugestão seria mapear quais as madeiras que são informadas com mais frequência, nas notas frias, para representar falsamente cada tipo de madeira

cuja extração está proibida, e, após essa etapa, construir classificadores especializados em discernir entre cada madeira, cuja extração é ilícita, e todos os seus “falsos representantes”, de maneira que, o agente alfandegário selecionaria o classificador que seria utilizado à depender da mercadoria declarada na nota fiscal.

Dessa forma, considere que a extração da espécie hipotética **X** está proibida, considere também, que as espécies hipotéticas que podem licitamente ser extraídas, e que são declaradas nas notas frias com o objetivo de burlar o tráfico de **X**, são: **A**, **B** e **C**. Nessa metáfora, **A**, **B** e **C** são os “falsos representantes” de **X**. Ainda no contexto desta metáfora, o agente alfandegário utilizaria, para esse caso específico, um classificador especializado em distinguir **X** de **A**, **B** e **C**.

Essa abordagem também seria muito útil para contornar a limitação explanada na Seção 4.3.2.1, que informa que a performance do classificador, construído à partir do modelo desenvolvido neste trabalho, se mostra quase sempre acima de 90% quando o universo de respostas possíveis é menor ou igual a 10, como pode ser constatado no gráfico 4.5, e, principalmente, no gráfico 4.6.

Neste contexto (entendendo que as espécies visadas pelo tráfico ilegal não são muito numerosas) equipar os agentes alfandegários com vários classificadores (um para cada espécie que deve ser monitorada), e orientá-los para utilizar cada classificador em função das notas fiscais apresentadas, parece uma alternativa promissora.

Referências Bibliográficas

ABDI, H. *Normalizing Data*. [S.l.], 2010. 4 p.

AGATHOS, A. *Surface reconstruction and simplification*. 87 p. Dissertação (Mestrado) — National and Kapodistrian University of Athens, 2008.

Bessa, F. M. S. *Criação de uma Xiloteca eletrônica (e-xiloteca) tropical e sua utilização para identificação e caracterização de Madeiras com fins científicos e econômicos*. 370 p. Tese (Doutorado) — Universidade Técnica de Lisboa, 2009.

BREMANANTH, R.; NITHYA, B.; SAIPRIYA, R. Wood Species Recognition System. In: *World Academy of Science, Engineering and Technology*. [S.l.: s.n.], 2009. p. 873–879.

CARRERO, G. C. *Dinâmica do desmatamento e consolidação de propriedades rurais na fronteira de expansão agropecuária*. 79 p. Dissertação (Mestrado) — INPA, 2009.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. *Semi-Supervised Learning*. MIT Press, 2006. 508 p. ISBN 9780262033589. Disponível em: <<http://www.bibsonomy.org/bibtex-/27897c1bbec1b3a621e660cd875ca101b/christine111>>.

DEITEL, H. M.; DEITEL, P. J. *Java: Como Programar*. 6. ed. [S.l.]: Pearson Prentice Hall, 2005. 1110 p. ISBN 978-85-7605-019-3.

DIETRICH, T. G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. In: *Neural Computation 10*. [S.l.: s.n.], 1998. p. 1895–1923.

DUAN, K.; KEERTHI, S. S.; POO, A. N. Evaluation of simple performance measures for tuning SVM hyperparameters. In: *Helsinki University of Technology, Laboratory of Computational Engineering publications*. [S.l.: s.n.], 2002.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. 2.. ed. New York: Wiley-Interscience Publication, 2001. 654 p. Disponível em: <<http://www.bibsonomy.org-/bibtex/25ef4fe4778daaf4b4e56c0d66161e048/flint63>>.

Empresa, EMBRAPA. dez. 1972.

FELIPE, C. J. *Desenvolvimento de métodos para extração, comparação e análise de características intrínsecas de imagens médicas, visando à recuperação perceptual por conteúdo*. 176 p. Tese (Doutorado) — USP, 2005.

FERREIRA, A. A. *Comparação de Arquiteturas de redes neurais para sistemas de reconhecimento de padrões em narizes artificiais*. 119 p. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2004.

FERRERO, C. A. *Algoritmo KNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia*. 129 p. Dissertação (Mestrado) — USP, jan. 2009.

FREUND, J. E.; SIMON, G. A. *Estatística Aplicada: economia, administração e contabilidade*. 9. ed. [S.l.]: Bookman, 2000. 404 p. ISBN 8573075317.

GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo: Pearson Prentice-Hall, 2010. 624 p. ISBN 9788576054016.

HEDAYATI, F.; CHONG, J.; KEUTZER, K. *Recognition of Tibetan Wood Block Prints with Generalized Hidden Markov and Kernelized Modified Quadratic Distance Function*. Dissertação (Mestrado) — EECS Department, University of California, Berkeley, Nov 2010. Disponível em: <<http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-139.html>>.

JAIN, R. C.; KASTURI, R.; SCHUNCK, B. G. *Machine vision*. McGraw-Hill, 1995. I–XX, 1–549 p. ISBN 978-0-07-032018-5. Disponível em: <<http://www.bibsonomy.org/bibtex-/251f0aefdda2612e0c19490f06b58c36a/dblp>>.

JÚNIOR, C. M. d. S. Avanços do sensoriamento remoto para o monitoramento da exploração madeireira na Amazônia. In: *Anais XIII Simpósio Brasileiro de Sensoriamento Remoto*. [S.l.: s.n.], 2007. p. 6987–6994.

KAUPPINEN, H. *Development of a color machine vision method for wood surface inspection*. Tese (Doutorado), 1999. Dissertation. Acta Univ Oul C 141, 138 p + App. Disponível em: <<http://herkules.oulu.fi/isbn9514254244>>.

KHALID, M. et al. *Design of an intelligent wood species recognition system*. [S.l.], 2008. 10 p.

KOHAVI, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: . [S.l.]: Morgan Kaufmann, 1995. p. 1137–1143.

LOANNIDIS, Y. The history of histograms (abridged). In: *VLDB Conference*. [S.l.: s.n.], 2003. p. 124.

MAENPAA, T. *The Local Binary Pattern Approach to Texture Analysis – Extensions and Applications*. 80 p. Tese (Doutorado) — University of Oulu, 2003.

MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. 414 p. ISBN 978-0-07-042807-2. Disponível em: <<http://www.bibsonomy.org/bibtex-/23e79734ee1a6e49aee02ffd108224d1c/flint63>>.

NASIRZADEH, M. Woods Recognition System Based on Local Binary Pattern. In: *Computational Intelligence, Communication Systems and Networks (CICSyN)*. [S.l.: s.n.], 2010. p. 308–313.

NISKANEN, M. *A visual training based approach to surface inspection*. 132 p. Dissertação (Mestrado) — University of Oulu, 2003.

NISKANEN, M.; SILVEN, O.; KAUPPINEN, H. *Color and texture based wood inspection with non-supervised clustering*. [S.l.], 2009. 7 p.

ODOM, D. *Differential pixel statistic based priors for image restoration*. 97 p. Dissertação (Mestrado) — University of California, jun. 2006.

PAN, S.; KUDO, M. Recognition of Wood Porosity Based on Direction Insensitive Feature Sets. In: *Transactions on Machine Learning and Data Mining*. [S.l.: s.n.], 2012. v. 5, p. 45–62. ISBN 978-3-942952-11-8.

PETRAUSKI, S. M. F. et al. Competitividade do Brasil no mercado internacional de madeira serrada. In: *Cerne*. [S.l.: s.n.], 2011. v. 18, p. 99–104.

PETROU, M.; SEVILLA, P. G. *Image processing - dealing with texture*. Wiley, 2006. 618 p. ISBN 978-0-470-02628-1. Disponível em: <<http://www.bibsonomy.org/bibtex-/2767f938cb0cff170ba7cb642b504e059/dblp>>.

PIETIKÄINEN, M. et al. *Computer Vision Using Local Binary Patterns*. [S.l.]: Springer, 2011. 224 p. ISBN 978-0-85729-747-1.

- PUTTER, T. v. d. *Using the discrete 3D Voronoi diagram for the modelling of 3D continuous information in geosciences*. 124 p. Dissertação (Mestrado) — Utrecht University, 2010.
- QIU, Z. F. *A Simple Machine Vision System for Improving the Edging and Trimming Operations Performed in Hardwood Sawmills*. 86 p. Dissertação (Mestrado) — Faculty of the Virginia Polytechnic Institute and State University, 1999.
- REFAEILZADEH, P.; TANG, L.; LIU, H. *Cross-Validation*. [S.l.], nov. 2008.
- ROSENBERG, C.; HEBERT, M.; THRUN, S. *Color Constancy Using KL-Divergence*. [S.l.], 2007. 80 p.
- SALZBERG, S. L. On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. In: *Data Mining and Knowledge Discovery*. [S.l.: s.n.], 1997. p. 317–328.
- SANTOS, R. P. *Os principais fatores do desmatamento na Amazônia (2002-2007) - uma análise econométrica e espacial*. 129 p. Dissertação (Mestrado) — Universidade de Brasília, 2010.
- SONKA, M.; HLAVAC, V.; BOYLE, R. *Image Processing, Analysis, and Machine Vision*. 3. ed. Thomson-Engineering, 2007. 829 p. Disponível em: <<http://www.bibsonomy.org/bibtex/219bc5dc0249a5fb9f2179f22ab764178/qmerigot>>.
- SYED, A. R. *A Review of Cross Validation and Adaptive Model Selection*. 99 p. Dissertação (Mestrado) — Georgia State University, 2011.
- TAY, K. S. *Improving Digital Ink Interpretation through Expected Type Prediction and Dynamic Dispatch*. Dissertação (Mestrado) — Massachusetts Institute of Technology, maio 2008.
- TEO, C. K. *Digital enhancement of night vision and thermal images*. 93 p. Dissertação (Mestrado) — NAVAL POSTGRADUATE SCHOOL, dez. 2003.
- TOU, J. Y.; LAU, P. Y.; TAY, Y. H. *Computer Vision-based Wood Recognition System*. [S.l.], 2008.
- TOU, J. Y.; TAY, Y. H.; LAU, P. Y. A Comparative Study for Texture Classification Techniques on Wood Species Recognition Problem. In: *Fifth International Conference on Natural Computation*. [S.l.: s.n.], 2009. p. 8–12.

TOU, J. Y.; TAY, Y. H.; LAU, P. Y. Rotational Invariant Wood Species Recognition through Wood Species Verification. In: *First Asian Conference on Intelligent Information and Database Systems*. [S.l.: s.n.], 2009.

VIEIRA, R. T. et al. *Uma nova proposta para análise de textura local por conjuntos fuzzy*. [S.l.], 2012.

WANG, B.-h. Wood recognition based on grey-level co-occurrence matrix. In: *Computer Application and System Modeling (ICCASM)*. [S.l.: s.n.], 2010. p. 269–272.

ZHONG, D. *Image Database Retrieval Methods Based on Feature Histograms*. Tese (Doutorado) — Tampere University of Technology, maio 2008.

APÊNDICE A

RESULTADOS DA QUINTA BATERIA DE TESTES

Conforme explanado anteriormente, os testes da quinta bateria são os únicos nos quais as imagens da Embrapa e da UFPR foram mescladas. Desta forma, foi possível testar a performance do algoritmo, considerando um banco de imagens heterogêneo (com relação à resolução(tamanho) e ao formato).

As imagens da UFPR foram disponibilizadas através da base *Forest Species Database*. O algoritmo de classificação, considerado nesta bateria de testes, foi o *KNN*. A tabela que se segue descreve todos os resultados alusivos à quinta bateria de testes.

Tabela A.1: Resultados da quinta bateria de testes (K-Fold Cross-Validation)

GI-GF (Qtd.)	Espécies	Distância	Normalização	Média de Acerto (%)	Desvio Padrão	Coef. de Variação (%)
1-9 (63)	9	Euclidiana	Não	90	12.76	14.18
1-9 (63)	9	Euclidiana	Sim	91,66	12.62	13.76
1-9 (63)	9	KLD	Não	93,33	13.33	14.28
1-9 (63)	9	KLD	Sim	93,33	13.33	14.28
1-10 (83)	10	Euclidiana	Não	92,5	9.57	10.35
1-10 (83)	10	Euclidiana	Sim	93,75	9.46	10.09
1-10 (83)	10	KLD	Não	95	10	10.52
1-10 (83)	10	KLD	Sim	95	10	10.52
1-11 (103)	11	Euclidiana	Não	93	8.24	8.86
1-11 (103)	11	Euclidiana	Sim	94	7.65	8.14
1-11 (103)	11	KLD	Não	96	8	8.33
1-11 (103)	11	KLD	Sim	96	8	8.33
1-12 (123)	12	Euclidiana	Não	90,93	7.44	8.18
1-12 (123)	12	Euclidiana	Sim	91,77	6.42	6.99
1-12 (123)	12	KLD	Não	90,96	5.79	6.36
1-12 (123)	12	KLD	Sim	90,96	5.79	6.36
1-13 (143)	13	Euclidiana	Não	92,21	6.37	6.91
1-13 (143)	13	Euclidiana	Sim	92,93	6.47	6.97
1-13 (143)	13	KLD	Não	92,23	4.95	5.37

Continua na próxima página

GI-GF (Qtd.)	Espécies	Distância	Normalização	Média de Acerto (%)	Desvio Padrão	Coef. de Variação (%)
1-13 (143)	13	KLD	Sim	92,23	4.95	5.37
1-14 (163)	14	Euclidiana	Não	91,38	4.31	4.72
1-14 (163)	14	Euclidiana	Sim	92,01	3.70	4.02
1-14 (163)	14	KLD	Não	90,79	2.40	2.65
1-14 (163)	14	KLD	Sim	90,79	2.40	2.65
1-15 (183)	15	Euclidiana	Não	90,11	4.69	5.21
1-15 (183)	15	Euclidiana	Sim	90,67	4.63	5.11
1-15 (183)	15	KLD	Não	89,58	4.99	5.57
1-15 (183)	15	KLD	Sim	89,58	4.99	5.57
1-16 (203)	16	Euclidiana	Não	89,62	4.18	4.67
1-16 (203)	16	Euclidiana	Sim	90,12	4.36	4.86
1-16 (203)	16	KLD	Não	90,61	4.48	4.95
1-16 (203)	16	KLD	Sim	90,61	4.48	4.95
1-17 (223)	17	Euclidiana	Não	88,78	2.70	3.04
1-17 (223)	17	Euclidiana	Sim	89,24	3.27	3.67
1-17 (223)	17	KLD	Não	90,56	4.82	5.33
1-17 (223)	17	KLD	Sim	90,56	4.82	5.33
1-18 (243)	18	Euclidiana	Não	88,49	2.26	2.55
1-18 (243)	18	Euclidiana	Sim	88,90	3.05	3.43
1-18 (243)	18	KLD	Não	90,11	4.31	4.79
1-18 (243)	18	KLD	Sim	90,11	4.31	4.79
1-19 (263)	19	Euclidiana	Não	88,21	3.15	3.57
1-19 (263)	19	Euclidiana	Sim	88,60	3.83	4.33

Continua na próxima página

GI-GF (Qtd.)	Espécies	Distância	Normalização	Média de Acerto (%)	Desvio Padrão	Coef. de Variação (%)
1-19 (263)	19	KLD	Não	89,33	4.89	5.48
1-19 (263)	19	KLD	Sim	89,33	4.89	5.48
1-20 (283)	20	Euclidiana	Não	86,57	4.27	4.93
1-20 (283)	20	Euclidiana	Sim	86,57	5.24	6.05
1-20 (283)	20	KLD	Não	86,67	5.10	5.75
1-20 (283)	20	KLD	Sim	88,67	5.10	5.75

Legenda/instruções para entendimento da tabela A.1:

GI-GF (Qtd.): Grupo Inicial - Grupo Final (Quantidade de Imagens)

Espécies: Indica a quantidade de espécies existentes no teste

KLD: Algoritmo de distância *Kullback Leibler-Divergence* original

Euclidiana: Algoritmo de distância euclidiana

Tabela A.2: Resultados da quinta bateria de testes (*Leave-One-Out Cross-Validation*)

GI-GF (Qtd.)	Espécies	Distância	Normalização	Acerto (%)
1-9 (63)	9	Euclidiana	Não	100
1-9 (63)	9	Euclidiana	Sim	100
1-9 (63)	9	KLD	Não	100
1-9 (63)	9	KLD	Sim	100
1-10 (83)	10	Euclidiana	Não	100
1-10 (83)	10	Euclidiana	Sim	100
1-10 (83)	10	KLD	Não	100
1-10 (83)	10	KLD	Sim	100
1-11 (103)	11	Euclidiana	Não	99,03
1-11 (103)	11	Euclidiana	Sim	99,03
1-11 (103)	11	KLD	Não	100
1-11 (103)	11	KLD	Sim	100
1-12 (123)	12	Euclidiana	Não	96,75
1-12 (123)	12	Euclidiana	Sim	96,75
1-12 (123)	12	KLD	Não	95,93
1-12 (123)	12	KLD	Sim	95,93
1-13 (143)	13	Euclidiana	Não	97,20
1-13 (143)	13	Euclidiana	Sim	97,20
1-13 (143)	13	KLD	Não	96,50

Continua na próxima página

GI-GF (Qtd.)	Espécies	Distância	Normalização	Acerto (%)
1-13 (143)	13	KLD	Sim	96,50
1-14 (163)	14	Euclidiana	Não	95,09
1-14 (163)	14	Euclidiana	Sim	95,09
1-14 (163)	14	KLD	Não	94,48
1-14 (163)	14	KLD	Sim	94,48
1-15 (183)	15	Euclidiana	Não	94,54
1-15 (183)	15	Euclidiana	Sim	94,54
1-15 (183)	15	KLD	Não	93,44
1-15 (183)	15	KLD	Sim	93,44
1-16 (203)	16	Euclidiana	Não	95,07
1-16 (203)	16	Euclidiana	Sim	95,07
1-16 (203)	16	KLD	Não	94,09
1-16 (203)	16	KLD	Sim	94,09
1-17 (223)	17	Euclidiana	Não	94,62
1-17 (223)	17	Euclidiana	Sim	94,62
1-17 (223)	17	KLD	Não	93,72
1-17 (223)	17	KLD	Sim	93,72
1-18 (243)	18	Euclidiana	Não	94,65
1-18 (243)	18	Euclidiana	Sim	94,65
1-18 (243)	18	KLD	Não	93,83
1-18 (243)	18	KLD	Sim	93,83
1-19 (263)	19	Euclidiana	Não	94,68
1-19 (263)	19	Euclidiana	Sim	94,68

Continua na próxima página

GI-GF (Qtd.)	Espécies	Distância	Normalização	Acerto (%)
1-19 (263)	19	KLD	Não	93,92
1-19 (263)	19	KLD	Sim	93,92
1-20 (283)	20	Euclidiana	Não	93,29
1-20 (283)	20	Euclidiana	Sim	93,29
1-20 (283)	20	KLD	Não	93,64
1-20 (283)	20	KLD	Sim	93,64

Legenda/instruções para entendimento da tabela A.2:

GI-GF (Qtd.): Grupo Inicial - Grupo Final (Quantidade de Imagens)

Espécies: Indica a quantidade de espécies existentes no teste

KLD: Algoritmo de distância *Kullback Leibler-Divergence* original

Euclidiana: Algoritmo de distância euclidiana

Os resultados das tabelas A.1 e A.2 foram sumarizados no gráfico 4.5.