REDUÇÃO DE ATRIBUTOS UTILIZANDO ANÁLISE DISCRIMINANTE COM APLICAÇÕES NA DETECÇÃO DE DEFEITOS EM COURO BOVINO

Willian Paraguassu Amorim

Dissertação de Mestrado apresentada ao
Departamento de Computação e Estatística do
Centro de Ciências Exatas e Tecnologia da
Universidade Federal de Mato Grosso do Sul

Orientador: Prof. Dr. Hemerson Pistori

O autor teve apoio financeiro da FINEP durante o desenvolvimento deste trabalho.

Agosto de 2009

REDUÇÃO DE ATRIBUTOS UTILIZANDO ANÁLISE DISCRIMINANTE COM APLICAÇÕES NA DETECÇÃO DE DEFEITOS EM COURO BOVINO

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Willian Paraguassu Amorim
e aprovada pela comissão julgadora.

Campo Grande/MS, 06 de julho de 2009.

Banca Examinadora:

- Prof. Dr. Cláudio Rosito Jung (UNISINOS)
- Profa. Dra. Maria Bernadete Zanusso (DCT-UFMS)
- Prof. Dr. Hemerson Pistori (orientador) (UCDB)

Agradecimentos

Agradeço a Deus, por iluminar meu caminho durante mais essa fase da minha vida, com muita proteção, paz e saúde.

Agradeço também a Nossa Senhora Aparecida, minha protetora de todos os dias.

Gostaria de agradecer também aos meus pais pela força e carinho. A toda minha família pelo incentivo e confiança. Aos meus amigos pelo apoio. Agradeço também a minha namorada Natália, pelas palavras de carinho, força e pela enorme paciência durante todos os momento difíceis.

Agradeço ao professor Dr. Hemerson Pistori, por novamente me apoiar, confiar no meu trabalho, e por todos os conselhos que me fizeram crescer pessoalmente e profissionalmente.

Obrigado a todos.

Resumo

Este trabalho apresenta um estudo e análise de técnicas de redução de atributos, baseada na análise discriminante aplicada a problemas de detecção de defeitos em imagens de couros bovinos no estágio couro cru e wet-blue. Foi realizado um estudo sobre casos que geram problemas no uso da análise discriminante quando aplicada em situações propícias a problemas de singularidade. Das soluções encontradas, FisherFaces, CLDA, DLDA, YLDA e a técnica de Kernel, implementamos cada uma, e realizamos experimentos de desempenho, analisando a taxa de acerto, tempos de treinamento e classificação, à medida que a quantidade de atributos é reduzida.

Os resultados experimentais indicaram que a redução de atributos pode manter a eficiência na classificação, mesmo em situações onde ocorre ou não a singularidade. Foram realizadas análises comparativas, apresentando cada resultado de desempenho comparados a técnicas de redução de atributos e classificadores diferentes. Identificamos também quais as melhores técnicas de extração de atributos e algoritmos de classificação, apresentando uma breve avaliação quanto a seus desempenhos e custo de processamento. E por fim realizamos uma comparação entre o sistema de classificação automática desenvolvido com a classificação feita manualmente por especialistas na área.

Palavras-chave: Visão Computacional, Análise Discriminante, Couro Bovino.

Abstract

TITLE: "ATTRIBUTES REDUCTION USING DISCRIMINANT ANALYSIS WITH APPLICATIONS IN BOVINE LEATHER DEFECTS DETECTION".

This paper presents a study and analysis of techniques for reduction of attributes, both based on discriminant analysis applied to problems of detection of defects in images taken from bovine leathers in raw and wet-blue stages. It has been done a study about problems that are likely to reach singularity problem. For this, we have implemented FisherFaces, CLDA, DLDA, YLDA, and Kernel technique algorithms. Then, we performed performance experiments in order to evaluate the classification rate as the number of attributes decrease.

The experimental results based on comparative analysis of different reduction and classification techniques indicated us that the attribute reduction provides great efficiency in classification process, even with the singularity problem. Additionally, we present the best feature extraction techniques applied to this problem, showing their performance and computational cost. Finally, we compared the automatic system classification developed here with a specialist-guided classification.

Keywords: Computer Vision, Discriminant Analisys, Leather.

Lista de Figuras

1.1 Exemplo de diagrama que ilustra a fase em que métodos de redução de atributos são utilizados em aplicações de Visão Computacional. 4

2.1 Exemplo de árvore de Decisão 11

2.2 Exemplo utilizando matriz de co-ocorrência. (a) imagem $M[i, j]$ com níveis de cinza de 0,1 e 2 . (b) matriz de co-ocorrência $c[i, j]$. 13

2.3 Exemplo utilizando mapas de interação. (a) Cálculo do mapa de interações. (b) Mapa polar de interações, onde I é a intensidade. 13

2.4 Representação gráfica do modelo RGB 14

2.5 Representação gráfica do modelo HSB 15

2.6 Exemplo da criação do histograma a partir de uma amostra de couro com defeito. (a) amostra com um risco de defeito, (b) histograma na componente R(Red) da amostra, (c) histograma na componente G(Green) da amostra, (d) histograma na componente B(Blue) da amostra. 15

2.7 Exemplo da seleção de um sub-conjunto X_M, a partir de um conjunto de atributos X_N. 17

2.8 Exemplo da utilização de seleção de atributos, utilizando duas classes e 2 atributos x e y. (a) conjunto de amostras, (b) projeção dos dados no atributo x e (c) projeção dos dados no atributo y 18

2.9 Exemplo da utilização da redução de atributos de um conjunto X_N, para a geração de um novo conjunto Y_M. 18

2.10 Exemplo da utilização da redução de atributos, utilizando duas classes e 2 atributos x e y. (a) conjunto de amostras, (b) geração do novo atributos e (c) projeção dos dados para o novo atributo gerado. 19

2.11 Exemplo da projeção de um conjunto de amostras sobre a componente principal. 20

2.12 Partes do Couro Bovino (Classificação Brasileira). (A) Cabeça, (B) Flanco, (C) Grupão 21

2.13 Imagens de couro bovino. (a) Imagem couro cru, (b) Imagem couro no estado Wet-Blue. 21

2.14 Imagem de defeitos de couros bovino nos processos (Couro Cru e Wet-Blue). (a) e (e) defeito sarna, (b) e (f) defeito carrapato, (c) e (g) defeito marca fogo, (d) e (h) defeito risco. 22

4.1 Exemplo utilizando Análise Discriminante de Fisher para redução de atributos. (a) conjunto de amostras utilizando 2 atributos x e y, (b) redução de atributos para um único atributo z 29

4.2 Exemplo da técnica de PCA aplicada na redução de atributos, (a) e (c) conjunto de amostras, (b) e (d) suas respectivas projeções. 30

4.3 Exemplo da Técnica de Fisher aplicada na redução de atributos, (a) e (c) conjunto de amostras, (b) e (d) suas respectivas projeções. 31

4.4 Amostras para a redução de atributos. (a), (c) e (e) Exemplo de amostras. (b), (d) e (f) Redução de seu respectivo conjunto de amostras. 41

5.1 (a) Tela de Aquisição de Imagens para marcação e rotulação e (b) Tela do gerador de amostras. 45

5.2 Tela do segundo módulo para a criação de bases de aprendizagem. 46

5.3 Modelo de interação dos quatro módulos implementados. 46

5.4 Tela para criação de regras para classificação. 47

6.1 Modelo projetado para captura das imagens. (a) tripé montato, (b) Posicionamento para captura das imagens. 49

6.2 Àreas para capturas de imagens. (a) iluminação artificial, (b) iluminação natual. 49

6.3 Exemplo de amostras semelhantes. (a) e (c) Amostras com defeito de Risco, (b) e (d) Amostras com defeito de esfola. 51

6.4 Exemplo de imagens marcadas manualmente pelo módulo de aquisição e marcação de imagens, com as seguintes cores, (vermelho = defeito, amarelo = fundo e verde = sem defeito). (a) defeito marca ferro, (b) defeito risco, (c) defeito sarna e (d) defeito carrapato. 52

6.5 Comportamento do classificador, (a) classificação correta(%), (b) tempo de treinamento em $log_2(s)$, (c) tempo de classificação em $log_2(s)$. 57

6.6 Comportamento do classificador, (a) classificação correta(%), (b) tempo de treinamento em $log_2(s)$, (c) tempo de classificação em $log_2(s)$. 60

6.7 Amostras visualmente semelhantes (a), (b), (c) defeito berne, (d), (e), (f) defeito carrapato, (g), (h), (i) defeito risco, (j), (k), (l) defeito esfola e (m), (n), (o) defeito furo. 61

6.8 Comportamento e Comparação de técnicas de Redução de Atributos para bases de aprendizagem que não ocorrem a singularidade no Couro Cru. 65

6.9 Comportamento e Comparação de técnicas de Redução de Atributos para bases de aprendizagem que não ocorrem a singularidade no Couro Wet-Blue. 66

6.10 Modelo de varredura para a classificação automática. 70

6.11 Imagens utilizadas para testes utilizando o módulo de classificação automática do DTCOURO. (a) couro wet-blue (b) couro cru. 71

6.12 Legenda de cores para a classificação automática. 72

6.13 Imagem Wet-Blue, utilizada para testes utilizando o módulo de classificação automática do DTCOURO. (a) Imagem Original, (b) Imagem classificada pelo sistema com menor resolução, (c) Imagem classificada pelo sistema com maior resolução, (d) Imagem classificada manualmente. 73

6.14 Imagem Couro-Cru, utilizada para testes utilizando o módulo de classificação automática do DTCOURO. (a) Imagem Original, (b) Imagem classificada pelo sistema com menor resolução, (c) Imagem classificada pelo sistema com maior resolução, (d) Imagem classificada manualmente. 74

Lista de Tabelas

2.1 Principais defeitos que comprometem a qualidade do couro bovino no Brasil . . . 22
2.2 Classificação do Couro Bovino no Brasil 23
2.3 Classificação do Couro Bovino nos EUA 23
2.4 Comparação entre receitas entre Brasil e os Estados Unidos - Wet blue (preço mercado internacional US$ 50.00) 23
2.5 Comparação entre receitas entre Brasil e os Estados Unidos - Semi Acabado (preço mercado internacional US$ 60.00) 23
2.6 Comparação entre receitas entre Brasil e os Estados Unidos - Acabado (preço mercado internacional US$ 80.00) 23

6.1 Atributos extraídos para o experimento de classificação de defeitos do couro-cru . 53
6.2 Parâmetros setados para o classificador C4.5 54
6.3 Parâmetros setados para o classificador IBk 54
6.4 Parâmetros setados para o classificador SMO 54
6.5 Parâmetros setados para o classificador NaiveBayes 54
6.6 Resultado classificação imagens Couro-Cru 55
6.7 Resultado classificação imagens Wet-Blue 56
6.8 Quantidade de Atributos Vs. Tempo de Classificação 58
6.9 Atributos extraídos segundo experimento 63
6.10 Resultado classificação(%) couro Wet-Blue 67
6.11 Resultado classificação(%) Couro Cru 67

Sumário

1 Introdução **3**

1.1 Justificativa 5

1.2 Objetivos 6

1.2.1 Geral 6

1.2.2 Específicos 6

1.3 Metodologia 7

1.4 Organização do Texto 8

2 Fundamentação Teórica **9**

2.1 Reconhecimento de Padrões 9

2.2 Aprendizagem de Máquina 10

2.2.1 Aprendizagem Supervisionada 10

2.2.2 Aprendizagem Não Supervisionada 11

2.3 Extração de Atributos 12

2.3.1 Matriz de Co-ocorrência 12

2.3.2 Mapas de Interação 13

2.3.3 Modelos de cores RGB, HSB e seus atributos 13

2.3.4 Filtro de Gabor 16

2.4 Seleção e Redução de Atributos 16

2.4.1 Análise de Componentes Principais - PCA 19

2.5 Couro Bovino 20

3 Trabalhos Correlatos **24**

3.1 Análise Discriminante 24

3.2 Detecção de Defeitos 26

4 Análise Discriminante de Fisher **27**

4.1 Redução de Atributos utilizando Fisher 27
4.2 FLDA Vs. PCA 28
4.3 Problemas Encontrados 30
4.4 Soluções Lineares 32
4.4.1 FisherFaces 32
4.4.2 Chen et al.'s Method (CLDA) 34
4.4.3 Yu and Yang's Method (DLDA) 34
4.4.4 Yang and Yang's Method (YLDA) 36
4.5 Soluções Não Lineares 37
4.5.1 Análise Discriminante Kernel 37
4.6 Exemplo Ilustrativo 39

5 Desenvolvimento **42**
5.1 Ferramentas de Apoio 42
5.1.1 IMAGEJ 43
5.1.2 WEKA 43
5.1.3 SIGUS 43
5.2 Ambiente de Aquisição e Marcação 43
5.3 Módulo de Geração de Experimentos 44
5.4 Módulo de Redução de Atributos e Classificação Automática 45
5.5 Módulo de Regras de Classificação 47

6 Experimentos, Resultados e Análise **48**
6.1 Equipamento 48
6.2 Procedimentos 48
6.3 Imagens utilizadas para os Experimentos 50
6.4 Marcação e Geração de Amostras 51
6.5 Classificação de defeitos do couro bovino no estágio Cru 52
6.5.1 Resultados e Análise 53
6.6 Classificação de defeitos do couro bovino no estágio Wet-Blue 55
6.6.1 Resultados e Análise 55
6.7 Redução de atributos sobre imagens com defeitos no Couro-Cru utilizando a técnica tradicional de Análise Discriminante de Fisher 56
6.7.1 Resultados e Análise 58

6.8 Redução de atributos sobre imagens com defeitos no Wet-Blue utilizando a técnica tradicional de Análise Discriminante de Fisher 58
6.8.1 Resultados e Análise 59
6.9 Experimento de simulação para o Problema de Singularidade 62
6.9.1 Resultados e Análise 63
6.10 Redução de atributos baseadas em LDA, utilizando imagens com defeitos sobre o Couro-Cru e Wet-Blue. 64
6.10.1 Experimento sobre as bases couro-cru e wet-blue sem problema de singularidade 64
6.10.2 Experimento sobre a base couro-cru e wet-blue com problema de singularidade 65
6.10.3 Análise dos Resultados 67
6.11 Classificação automática de defeitos em comparação à classificação por especialistas na área. 69
6.11.1 Módulo DTCOURO para classificação automática 69
6.11.2 Configuração do experimento 70
6.11.3 Resultado Classificação Sistema Vs. Especialista 72
6.11.4 Análise dos resultados 72

7 Conclusão **75**

A Conceitos de Álgebra **78**
A.1 Matriz 78
A.1.1 Determinante 79
A.1.2 Matriz Inversa 80
A.1.3 Matriz de Variância e Covariância 80
A.1.4 Auto-Valores e Auto-Vetores 80
A.1.5 Produto Escalar 81
A.1.6 Combinação Linear 82
A.1.7 Dependência e Independência Linear 83
A.1.8 Diagonalização de Matriz 83

Resumo

Este trabalho apresenta um estudo e análise de técnicas de redução de atributos, baseada na análise discriminante aplicada a problemas de detecção de defeitos em imagens de couros bovinos no estágio couro cru e wet-blue. Foi realizado um estudo sobre casos que geram problemas no uso da análise discriminante quando aplicada em situações propícias a problemas de singularidade. Das soluções encontradas, FisherFaces, CLDA, DLDA, YLDA e a técnica de Kernel, implementamos cada uma, e realizamos experimentos de desempenho, analisando a taxa de acerto, tempos de treinamento e classificação, à medida que a quantidade de atributos é reduzida.

Os resultados experimentais indicaram que a redução de atributos pode manter a eficiência na classificação, mesmo em situações onde ocorre ou não a singularidade. Foram realizadas análises comparativas, apresentando cada resultado de desempenho comparados a técnicas de redução de atributos e classificadores diferentes. Identificamos também quais as melhores técnicas de extração de atributos e algoritmos de classificação, apresentando uma breve avaliação quanto a seus desempenhos e custo de processamento. E por fim realizamos uma comparação entre o sistema de classificação automática desenvolvido com a classificação feita manualmente por especialistas na área.

Palavras-chave: Visão Computacional, Análise Discriminante, Couro Bovino.

Abstract

TITLE: "ATTRIBUTES REDUCTION USING DISCRIMINANT ANALYSIS WITH APPLICATIONS IN BOVINE LEATHER DEFECTS DETECTION".

This paper presents a study and analysis of techniques for reduction of attributes, both based on discriminant analysis applied to problems of detection of defects in images taken from bovine leathers in raw and wet-blue stages. It has been done a study about problems that are likely to reach singularity problem. For this, we have implemented FisherFaces, CLDA, DLDA, YLDA, and Kernel technique algorithms. Then, we performed performance experiments in order to evaluate the classification rate as the number of attributes decrease.

The experimental results based on comparative analysis of different reduction and classification techniques indicated us that the attribute reduction provides great efficiency in classification process, even with the singularity problem. Additionally, we present the best feature extraction techniques applied to this problem, showing their performance and computational cost. Finally, we compared the automatic system classification developed here with a specialist-guided classification.

Keywords: Computer Vision, Discriminant Analisys, Leather.

Capítulo 1

Introdução

A Visão Computacional é uma área que desenvolve métodos e técnicas que auxiliam na construção de sistemas computacionais capazes de realizar o processamento e a interpretação de imagens. Nos últimos anos, houve um grande aumento no desenvolvimento de sistemas de visão computacional, que trabalham para estabelecer um alto desempenho no reconhecimento automático de padrões e comportamentos em imagens digitais. O reconhecimento de padrões no contexto da visão computacional realiza um processo de classificação de imagens, a partir de informações ou características, previamente conhecidas, criando assim padrões de discriminalidade entre determinados grupos de classes.

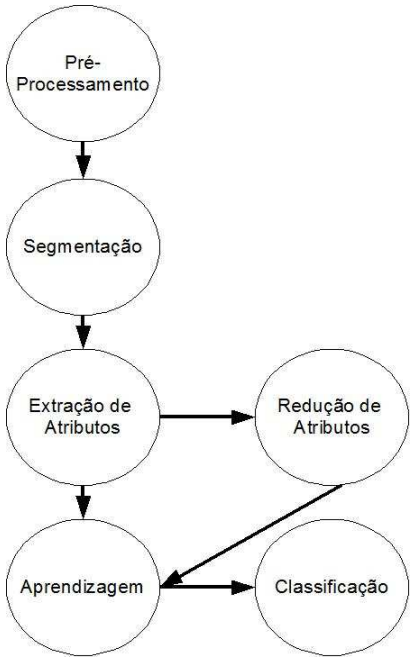
O Reconhecimento de Padrões é uma área desafiadora que ainda apresenta diversos problemas a serem resolvidos, quando aplicados utilizando técnicas de Visão Computacional. Junto a esses obstáculos, se encontra o problema de redimensionamento de atributos utilizando a técnica de Análise Discriminante de Fisher (FLDA) [36] [39].

Redimensionamento de atributos é uma sub-área do reconhecimento de padrões com objetivo de realizar a redução de um conjunto de atributos, gerando um novo sub-conjunto, que possa manter a eficiência na classificação das informações conhecidas. Uma das principais razões para o uso de uma técnica de redução de atributos é tentar diminuir ao máximo o custo de aprendizagem e maximizar a precisão do classificador automático sem que se perca a discriminalidade entre os padrões de classificação. Em problemas que possuem muitos atributos, após sua redução, a classificação será realizada de maneira mais rápida, contendo somente as informações mais discriminantes, e conseqüentemente ocupando menos memória.

A técnica de Análise Discriminante de Fisher é muito utilizada em aplicações que necessitam de uma redução de dimensão do seu espaço de atributos. O objetivo da técnica de Fisher é realizar uma combinação entre seus atributos, maximizando a razão da variância entre-grupos e inter-grupos e gerando novos espaços de atributos, que mais eficientemente separam suas classes. O Diagrama 1.1 ilustra em qual etapa a técnica de Análise Discriminante de Fisher ou qualquer outra técnica de redução de atributos pode ser aplicada, em problemas utilizando visão computacional.

A técnica de Fisher, por ser um modelo matemático que necessita de uma avaliação linear pode apresentar alguns problemas durante sua execução, entre eles se destaca o problema chamado de "matriz singular"[1]. Em alguns casos esse problema é encontrado quando realizamos

[1]Matriz Singular não admite inversa.



(a)

Figura 1.1: Exemplo de diagrama que ilustra a fase em que métodos de redução de atributos são utilizados em aplicações de Visão Computacional.

a redução de atributos por Fisher, em aplicações que possuem uma quantidade de atributos superior a quantidade de amostras para treinamento, resultando assim na singularidade. Para isso existem algumas regras em sua utilização que, sendo seguidas, poderão evitar, mas que na prática são difíceis de serem adotadas, como por exemplo: (1) as classes sob investigação devem ser mutuamente exclusivas, (2) cada classe deverá pertencer a uma população normal multivariada, (3) duas medidas não podem ser perfeitamente correlacionadas, entre outras.

Esse trabalho tem como objetivo apresentar uma análise do desempenho, em diferentes situações práticas, no uso da redução de atributos utilizando a técnica de análise discriminante de Fisher, e os principais problemas em sua utilização, com base em diversas soluções lineares e não lineares dadas na literatura. Os métodos utilizados em nosso trabalho foram: FisherFaces [3], Chen et al.'s (CLDA) [6], Yu and Yang's (DLDA) [36], Yang and Yang's (YLDA) [35] e a técnica baseada em Kernel KLDA [37]. Demonstramos também seus conceitos e desempenhos de cada técnica e realizamos breve comparação no uso de Fisher e Análise de Componentes Principais, destacando suas fundamentais diferenças e aplicações.

Para a utilização da redução de atributos por Fisher, utilizamos como informações de entrada para os algoritmos de classificação, diversas técnicas de extração de atributos, que são baseadas em cores, formas e textura. Essas técnicas são: matriz de co-ocorrência, mapas de interação, Filtros de Gabor, média HSB, média RGB e discretização de RGB e HSB. Na parte de aprendizagem e classificação, utilizamos algumas das técnicas mais utilizadas em reconhecimento de padrões, que são: árvores de decisão *(C4.5)*, máquina de vetores de suporte *(SVM)*, classificador bayesiano *(Naive Bayes)* e uma especialização de K-Vizinhos mais próximos *(IBk)*.

Através das técnicas implementadas realizamos diversos experimentos e análises no desempenho de classificação, utilizando como aplicação a detecção de defeitos em couro bovino. Esse tipo de aplicação é muito utilizado em problemas que necessitam de um resultado confiável e rápido para a detecção automática de padrões em algum tipo de textura. O objetivo da escolha desse problema é auxiliar o projeto DTCOURO com a parceria da Embrapa[2] na criação de um sistema automático para a detecção de defeitos em couro bovino.

Realizamos também experimentos utilizando a técnica tradicional de análise discriminante de Fisher e as técnicas implementadas como soluções para o problema de singularidade e destacamos as melhores e piores soluções, com base em taxa de classificação correta, tempo de treinamento e classificação, analisando assim a eficiência quanto a redução em problemas que ocorrem ou não a singularidade. Através de módulos implementados realizamos experimentos comparativos na classificação de imagens do couro bovino com imagens classificadas manualmente por um especialista na área.

Os resultados deste projeto estão ajudando a Embrapa na criação de um novo sistema de remuneração que possa valorizar a qualidade do couro bovino, a qual geralmente é citada como um dos principais motivos que inibem o aumento de investimentos na qualidade do couro bovino Brasileiro.

1.1 Justificativa

A redução de atributos é uma área de pesquisa desafiadora que abre inúmeras portas para aplicações que necessitam de eficiência no processo de classificação de imagens. A sua utilização aplicada na detecção de defeitos pode aumentar a velocidade no treinamento e na classificação automática.

Alguns trabalhos correlatos, como [1] e [5] apresentam a técnica de Análise Discriminante de Fisher como uma solução para o redimensionamento de atributos. Como resultados desses trabalhos a técnica de Fisher se comportou de maneira eficiente, sempre realizando a redução do espaço de atributos e mantendo a discriminalidade entre as classes de classificação. Mas em trabalhos como [3], [35] e [41] problemas de singularidade são encontrados no uso da técnica de Fisher, quando utilizada em aplicações que possuem uma quantidade alta de atributos e um conjunto pequeno de amostras.

A partir de uma parceira com a Embrapa Gado de Corte e o Grupo de Pesquisa em Engenharia e Computação (GPEC) foi projetada a criação de um sistema para detecção automática de defeitos em couro bovino. Essa aplicação servirá como apoio para a Embrapa na tomada de decisão quanto à criação de um novo modelo de sistema de remuneração do couro bovino, que possa contribuir com sua valorização em território nacional.

Com a realização desse trabalho estaremos também contribuindo, com novo estudo sobre a redução de atributos, em problemas de reconhecimento de padrões e classificação automática em texturas de alto nível de complexidade, como o couro bovino. E apresentaremos também de maneira inédita, resultados de classificação de couros bovinos, em estágios do couro cru e Wet-Blue, a partir de que, a quantidade de atributos para classificação é decrementada utilizando a técnica tradicional LDA e utilizando técnicas feitas para lidar com problemas que ocorrem à

[2] A Embrapa Gado de Corte, possui como objetivo transferir conhecimento tecnológico para o desenvolvimento sustentado do complexo produtivo nacional da carne bovina, visando a sua utilização e benefícios para a sociedade.

singularidade.

Com esse trabalho também poderemos mostrar o quanto é importante e eficiente o auxilio de sistemas de visão computacional, em que o objetivo esteja na busca pela otimização de serviços com qualidade. Com um sistema como este será possível ajudar especialistas da área, a melhorar a classificação do couro bovino e conseqüentemente aumentar sua qualidade, valorizando ainda mais o couro bovino em nosso país.

1.2 Objetivos

1.2.1 Geral

O objetivo desse trabalho foi a realização de um estudo da aplicação da Análise Discriminante de Fisher na redução de atributos baseados em textura, formas e cores para a detecção de defeitos em imagens digitais. O enfoque principal do trabalho esteve na análise do problema de redimensionamento de atributos por Fisher, em situações que ocorrem ou não a singularidade. A partir de estudos e experimentos, demonstramos problemas que ocorrem a singularidade seguido de diversas soluções dadas na literatura, que foram: CLDA, DLDA, FisherFace, YLDA, e a técnica de Kernel.

Com o objetivo de verificar a eficiência desses processos, foram realizados diversos experimentos e análise dos resultados da técnica de Fisher, na detecção automática de defeitos em couro bovino. Apresentamos também os principais métodos de extração de atributos, que foram utilizados na redução de sua dimensão, que são: matriz de co-ocorrência, mapas de interação, atributos de cores (HSB) e (RGB) e Filtros de Gabor.

Esse trabalho se concentrou na análise do comportamento da técnica de Fisher à medida que um conjunto de atributos é reduzido, demonstrando seus problemas com suas respectivas soluções e desempenhos computacionais. Realizamos também, testes no comportamento da técnica de Fisher, quanto a seu efeito no tempo de treinamento e tempo de classificação, utilizando os principais algoritmos de aprendizagem e classificação utilizados em projetos de visão computacional do grupo de pesquisa GPEC[3], que são: *C4.5*, *IBk*, *SMO* e *NaiveBayes*.

1.2.2 Específicos

1. Estudo dos principais conceitos relacionados à Análise Discriminante de Fisher;
2. Pesquisa das principais ferramentas existentes na área de visão computacional, que auxiliaram no desenvolvimento desse trabalho;
3. Implementação dos módulos de apoio para a realização dos experimentos;
4. Identificação dos principais problemas e propostas de soluções utilizando Análise Discriminante na redução de atributos;
5. Realização de experimentos na redução de atributos utilizando a técnica de Fisher e análise dos resultados;
6. Implementação das principais soluções do problema de singularidade da técnica de Fisher;

[3]Grupo de Pesquisa em Engenharia de Computação.

7. Realização de experimentos com base nas soluções implementadas e análise dos resultados;
8. Identificação das melhores soluções que envolvem a redução de atributos em problemas de detecção de defeitos em couros bovinos;
9. Identificação das melhores soluções que envolvem a redução de atributos em casos que ocorrem a singularidade;
10. Comparação dos resultados do sistema de classificação automática de couro bovino com a classificação realizada por especialista na área.
11. Análise dos resultados e conclusões.

1.3 Metodologia

Para o desenvolvimento desse trabalho, alguns conceitos e aplicações foram analisados e desenvolvidos de início. Como teoria realizamos diversas pesquisas sobre a análise Discriminante de Fisher e de trabalhos correlatos demonstrando sua diversas aplicações. Pesquisamos diversas ferramentas existentes na área de visão computacional, para auxiliar no desenvolvimento dos módulos utilizados nos experimentos, que são: Weka, ImageJ, Sigus e uma ferramenta de computação numérica conhecida como Scilab. Com isso implementamos os módulos de aquisição de imagens, módulo de geração de experimentos, módulo de redução de atributos e o módulo de classificação automática.

Com os módulos de apoio implementados focamos no estudo para identificar os principais problemas e propostas de soluções, utilizando Análise Discriminante de Fisher na redução de atributos. Inicialmente pesquisamos por trabalhos correlatos que apresentam os principais problemas da técnica LDA. Em seguida pesquisamos por trabalhos que apresentam soluções para o problema de singularidade da técnica LDA. A partir dessa pesquisa, conseguimos assim, identificar as principais técnicas utilizadas como soluções para problemas que ocorrem a singularidade, que foram as técnicas: FisherFace, Chen et al.'s (CLDA), Yu and Yang's (DLDA), Yang and Yang's (YLDA) e a técnica baseada em soluções não lineares conhecida como Kernel LDA(KLDA).

Através das pesquisas sobre possíveis soluções para problemas que envolvem a singularidade, foram realizadas as implementações de cada técnica. Inicialmente as técnicas foram implementadas utilizando uma linguagem de mais alto nível, conhecida como Scilab e em seguida utilizando a linguagem Java para a implementação das técnicas em formato de plugin para a ferramenta Sigus.

Com todas as técnicas e módulos implementados iniciamos o processo de captura de imagens do couro bovino. As imagens foram capturadas durante viagens junto a Embrapa e divididas no processo de couro-cru e wet-blue. Com isso criamos um banco de imagens com vários tipos padrões de defeitos e organizamos no sistema de tal forma que cada imagem pudesse ter regiões marcadas e rotuladas, identificando sua origem, processo, tipo de marcação e tipo de defeito.

Dessa forma conseguimos iniciar a fase de experimentos. Primeiramente foram realizados experimentos para medição do desempenho na classificação de imagens do couro-cru e Wet-Blue. Em seguida partimos para experimentos de redução utilizando a técnica tradicional LDA. Com base nessas primeiras respostas, realizamos experimentos utilizando as soluções pesquisadas

para o problema de singularidade e realizadas assim análise de desempenho de cada técnica implementada.

E por fim realizamos experimentos comparativos entre o resultado de classificação pelo sistema, com um especialista treinado na área, para analisar seu comportamento e identificar possíveis melhorias no sistema. Com todos esses resultados conseguimos realizar uma análise do desempenho de todas as técnicas aqui apresentadas identificando quais as melhores soluções para problemas que envolvem a singularidade, na redução de atributos aplicada na detecção de defeitos em couros bovinos.

1.4 Organização do Texto

O texto desse trabalho está dividido da seguinte forma. O Capítulo 2 oferece uma breve abordagem dos principais conceitos sobre reconhecimento de padrões, processamento digital de imagens, métodos de extração de atributos, seleção e a redução de atributos, análise de componentes principais e um breve estudo sobre o couro bovino. No Capítulo 3 são apresentados diversos trabalhos correlatos utilizando Fisher e detecção de defeitos. O Capítulo 4 apresenta a definição e teoria da técnica de Análise Discriminante de Fisher, aplicada na redução de atributos, seguido dos principais problemas e soluções. O Capítulo 5 demonstra todos os passos e ferramentas que foram utilizadas e integradas para a implementação dos módulos responsáveis pela realização dos experimentos. O Capítulo 6 apresenta os experimentos e resultados obtidos a partir dos módulos desenvolvidos, e por fim, o Capítulo 7 apresenta as conclusões.

Capítulo 2

Fundamentação Teórica

Os conceitos apresentados nessa seção, servem de base para o entendimento do trabalho aqui desenvolvido. Como modo de organização, a primeira seção apresenta a área de reconhecimento de padrões. Em seguida apresentamos conceitos de aprendizagem de máquina e suas aplicações. Nas seções seguintes, são apresentados os métodos de extração de atributos utilizados e de maneira conceitual e ilustrativa apresentamos as principais diferenças relacionadas à seleção e redução de atributos, utilizando como método principal a técnica de Análise de Componentes Principais. E por fim, nesse capítulo são apresentadas diversas considerações sobre o couro bovino e a visão da sua qualidade no Brasil.

Também podem ser encontrados no Anexo A, os principais conceitos elementares sobre álgebra, relacionados ao entendimento da redução de atributos.

2.1 Reconhecimento de Padrões

Reconhecimentos de Padrões é basicamente o processo de identificar uma categoria de padrões, baseada em um conjunto de dados. Esta área abrange desde a detecção de padrões até a escolha entre objetos para a classificação baseada em um conjunto de informações [32]. O reconhecimento de padrões é uma área de suma importância no cotidiano humano, devido às situações na vida humana tomar formas de padrões, como exemplo: formação da linguagem, modo de falar, desenho de figuras, entendimento de imagens, tudo se torna um padrão. Por ser de alta complexidade, na área de reconhecimento de padrões em imagens, existem diversos processos computacionais, que sendo utilizados, facilitam seu processo de reconhecimento, como por exemplo, a utilização de pré-processamento, extração e seleção de atributos, entre outras.

A área de reconhecimentos de padrões envolve inúmeros problemas de processamento de informações, indo desde reconhecimento de voz, detecção de erros em equipamentos ou até mesmo, inspeção visual para detecção de imperfeições em texturas. Na prática, algumas áreas que vem sendo muito utilizadas no reconhecimento de padrões [33], podem ser citadas:

1. Reconhecimento de Faces;
2. Identificação de impressões digitais;
3. Segmentação e pré-processamento de imagens;

4. Reconhecimento de Voz;
5. Detecção de movimentos;
6. Detecção de defeitos;

O reconhecimento de padrões tem como objetivo principal a distinção de diferentes tipos de padrões, criando formas ou regras para classificação entre classes de informações. Para entendermos melhor podemos apresentar o seguinte exemplo: dado um conjunto c de classes $c_1, c_2, ..., c_n$, e dado um padrão w, o reconhecimento do padrão será responsável em associar o padrão w a uma determinada classe do conjunto c [32].

Para conseguirmos obter as "regras" para classificação dessas informações é preciso utilizar técnicas de aprendizagem, ou seja, utilizar um conjunto de amostras de padrões para ensinar um determinado classificador. Para entendermos melhor, na seção a seguir serão apresentados os principais conceitos sobre aprendizagem de máquina, e diferenças entre aprendizagem supervisionada e não supervisionada.

2.2 Aprendizagem de Máquina

A aprendizagem de máquina pode ser considerada um sub-campo da inteligência artificial dedicada ao desenvolvimento de técnicas que permitem ao computador aprender e aperfeiçoar seu desempenho em alguma determinada tarefa. Em um nível mais geral, a aprendizagem pode ser imaginada como um agente, que contém um elemento de desempenho que decide que ações executar junto com um elemento de aprendizagem que modifica o elemento de desempenho para que ele tome as melhores decisões [26].

Em alguns casos, a aprendizagem pode variar desde uma memorização trivial de experiências, até mesmo na criação de regras específicas. Para entendermos melhor cada caso, apresentaremos nessa seção, alguns campos da aprendizagem de máquina, que são muito utilizados em trabalhos de reconhecimento de padrões, que são: aprendizagem supervisionada e não supervisionada.

2.2.1 Aprendizagem Supervisionada

O problema da aprendizagem supervisionada envolve a aprendizagem de uma função a partir de exemplos de suas entradas e saídas. Grande parte dos algoritmos dessa área toma como entrada um objeto ou situação descrita por um conjunto de atributos retornando assim uma decisão, ou o valor da saída, de acordo com a entrada. Estes algoritmos são preditivos, isso significa que suas tarefas desempenham inferências nos dados com o intuito de fornecer previsões ou tendências, obtendo informações não disponíveis a partir dos dados disponíveis. Em nossos experimentos aqui apresentados utilizamos a aprendizagem supervisionada para a classificação de imagens do couro bovino, depois que a redução ou não de atributos é realizada[30].

Com os algoritmos de aprendizagem e classificação supervisionada é possível determinar o valor de um atributo através dos valores de um subconjunto dos demais atributos da base de dados. Por exemplo, num conjunto de dados, deseja-se descobrir qual a classificação das imagens com textura listradas e textura quadriculadas. Com classificadores, pode-se inferir (prever) que as texturas com um atributo x, se possuir valor abaixo de y são texturas listradas, e acima desse valor, são texturas quadriculadas.

Neste caso, as informações como $(textura - listrada, textura - quadriculada)$ são denominadas classes, e a informação para a discriminação entre as classes, o atributo x, é considerado atributo das classes. As formas mais comuns de representação de conhecimento dos algoritmos de classificação são redes neurais, ou regras utilizando árvore de decisão, como ilustrado na Figura 2.1. Os algoritmos Id3, C45 ou J48, por exemplo, geram como resultado árvores de classificação, enquanto que outros como Prism, Part, OneR geram regras de classificação. Modelos matemáticos de regressão e redes neurais também representam resultados para a classificação, como também os algoritmos SMO, LinearRegression, Neural, entre outros.

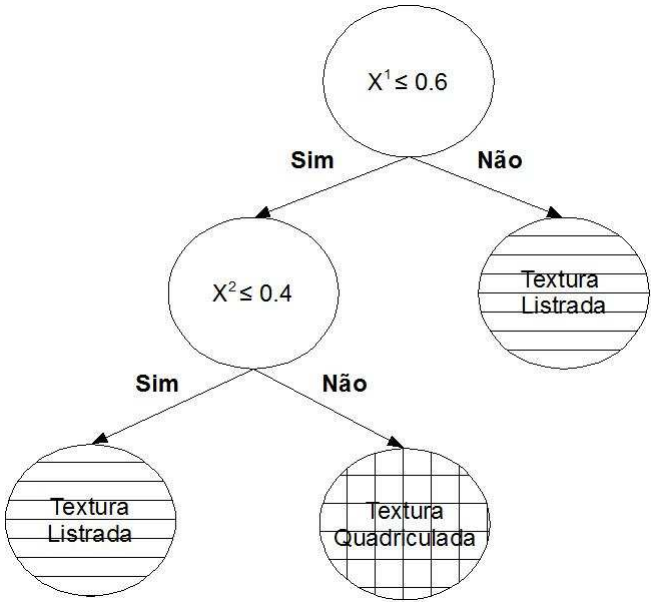


Figura 2.1: Exemplo de árvore de Decisão

Na Figura 2.1, a árvore de decisão alcança sua decisão, executando uma seqüência de passos e testes, em que cada nó da árvore corresponde a um teste a ser executado, e as ramificações, a partir do nó da árvore, são identificadas com os valores possíveis do teste. O nó da árvore é referente ao valor a ser retornado, caso aquela folha seja alcançada. A grande vantagem de árvore de decisão como meio de explicar sobre aprendizagem supervisionada, está em sua fácil representação como regras do tipo "se, senão", o que sendo mais aparentemente natural ao ser humano.

2.2.2 Aprendizagem Não Supervisionada

Nestes algoritmos o rótulo da classe de cada amostra do treinamento não é conhecida, e o número ou conjunto de classes a ser treinado pode não ser conhecido a priori, daí o fato de ser uma aprendizagem não-supervisionada. Além disso, são também descritivos, pois descrevem de forma concisa os dados disponíveis, fornecendo características das propriedades gerais dos dados.

Em outros termos podemos considerar a aprendizagem não supervisionada, como uma aprendizagem de padrões na entrada, quando não são fornecidos valores de saídas específicas. Esse tipo de aprendizagem não aprende diretamente que fazer, por que não tem nenhuma

informação sobre o que constitui uma ação correta ou uma ação desejável.

Esses tipos de algoritmos não foram utilizados em nossos experimentos, pelo fato de não utilizar informações relacionadas às classes de classificação. Sem esse tipo de informação fica complexa a avaliação de cada técnica de redução de atributos.

2.3 Extração de Atributos

Extrair as características (atributos) mais importantes numa imagem pode evidenciar as diferenças e similaridades entre os objetos. Algumas características são definidas pela aparência visual na imagem como: o brilho de uma determinada região, textura, amplitude do histograma, entre outras. O principal objetivo da extração de atributos é realizar uma combinação entre o conjunto de informações fornecidas, criando um espaço de atributos que melhor representam sua discriminabilidade.

A seguir serão apresentadas as técnicas de extração de atributos, que foram utilizadas como base para a realização dos experimentos dessa pesquisa, que são: matriz de co-ocorrência, mapas de interação, atributos de cores HSB e RGB e filtros de Gabor.

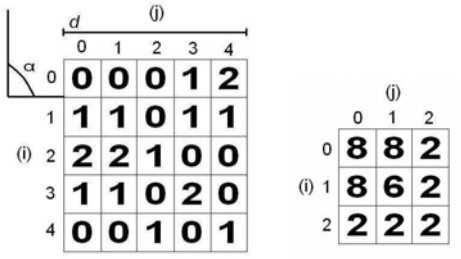
2.3.1 Matriz de Co-ocorrência

Matriz de co-ocorrência é uma técnica de extração de atributos que consiste em calcular a quantidade de combinações diferentes de valores de intensidade dos pixels de uma imagem. Sua função principal é tentar descrever a textura de uma imagem, através de um conjunto de valores, indicando a ocorrência, dos seus pixels, em níveis de cinza, para diferentes direções e diferentes ângulos.

A matriz de co-ocorrência de níveis de cinza fornece relações espaciais entre os pixels de uma imagem, possibilitando a extração de atributos para a representação de suas características de textura, como exemplo: rugosidade, granulosidade, aspereza, regularidade, direcionalidade, entre outras [19].

O cálculo da matriz de co-ocorrência é realizado a partir dos índices das linhas e colunas que representam os diferentes valores de níveis de cinza. Dessa forma calcula a freqüência que os mesmos ocorrem dois a dois em certa direção e distância [7]. As direções usualmente usadas em trabalhos correlatos são de 0^o, 10^o, 45^o, 90^o e 135^o e as distâncias são escolhidas de acordo com a granularidade das imagens manipuladas [29]. Com esses parâmetros é calculada a matriz de co-ocorrência para certa direção e distância.

Como exemplo, seja uma imagem representada por uma matriz M de pontos mostrada na Figura 2.3(a), onde cada valor da matriz M(i,j) representa a intensidade do pixel variando de 0 a 2. Nesse exemplo são extraídas informações sobre a relação dos pares de pixels com parâmetros de distância (d) 1 pixel e ângulo(α) de 0^o. Teremos assim como resultado a matriz de co-ocorrência mostrada na Figura 2.3(b). À medida que os pares de pixels são analisados sua relação é incrementada na matriz de co-ocorrência $c(i, j)$.

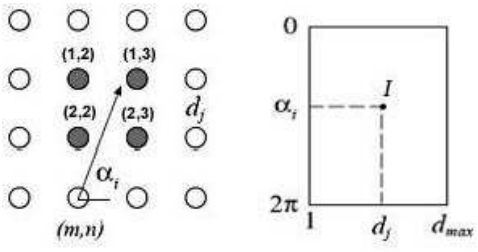


(a) (b)

Figura 2.2: Exemplo utilizando matriz de co-ocorrência. (a) imagem $M[i, j]$ com níveis de cinza de 0,1 e 2 . (b) matriz de co-ocorrência $c[i, j]$.

2.3.2 Mapas de Interação

Mapas de interação são muito utilizados na análise de pares de pixel de uma imagem [15]. Essa técnica consiste em calcular a diferença de valores das intensidades dos pixels, localizados em uma determinada distância e ângulo da imagem levando em consideração suas variações de textura e formas. Como a matriz de co-ocorrência, o mapa de interação também trabalha com a intensidade em tons de cinza dos pixels de uma imagem, mas com a diferença que sua intensidade é o resultado do cálculo da interpolação de seus pixels vizinhos. As Figuras 2.3(a) e 2.3(b) ilustram como é aplicada essa técnica na análise de uma imagem.



(a) (b)

Figura 2.3: Exemplo utilizando mapas de interação. (a) Cálculo do mapa de interações. (b) Mapa polar de interações, onde I é a intensidade.

Como é ilustrado na Figura 2.3(a), os píxels da imagem são analisados por (m, n), em posições dadas por um ângulo α e distância d. A intensidade é calculada no ponto (α_i, d_i) interpolando os píxels vizinhos como ilustrado na Figura 2.3(b). Com isso é gerado o mapa de interações polares $map(i, j)$, diferente das matrizes de co-ocorrência onde a intensidade é o valor do píxel. A interação desse mapa com os valores originais resulta em similaridades e diferenças, assim como nas matrizes de co-ocorrência.

2.3.3 Modelos de cores RGB, HSB e seus atributos

Em tudo que observamos, a cor esta presente, tornando-se assim um elemento essencial em visão computacional. A cor transmite informações que dão realismo e naturalidade nas imagens,

como condições de iluminação e a forma dos objetos visualizados. Para ter uma representação original da cor devemos utilizar modelos de cores, de forma correta e precisa.

A cor pode ser formada por três tipos de processos: (1) processos aditivos, quando há ocorrência de dois ou mais raios luminosos de freqüência diferentes, (2) processo por subtração, quando a luz é transmitida de um filtro que absorve radiação luminosa de um determinado comprimento de onda, ou pelo processo por (3) formação de pigmentação, que ocorre quando os pigmentos absorvem refletem ou transmitem a radiação luminosa [8].

As cores podem ser divididas em três grupos distintos que são: cores primárias, cores secundárias e cores terciárias. Cores primárias são formadas por meios das cores básicas vermelho, verde e azul (RGB). Para a criação das cores secundárias são utilizadas as combinações de duas cores primárias. E para a representação das cores terciárias são obtidos através da mistura de uma cor primária com uma ou mais cores secundárias.

Diferentes maneiras podem ser utilizadas para a representação das cores de uma imagem no computador. São muitos os espaços de cores existentes, e entre eles podemos citar o RGB, HSB, CMY, YIQ, YUV, HSI, HSV, CIELAB, CIELUV, rgb, $c_1c_2c_3$, $l_1l_2l_3$, YCb Cr. Nesse trabalho apresentaremos somente duas dessas formas de representação de cores, que serão muito utilizadas em nossos experimentos, que são: RGB e HSB.

Modelo RGB

O modelo RGB é um modelo de cor com três cores primárias: vermelho, verde e azul. A sigla RGB deriva da junção das primeiras letras dos nomes das cores primárias em língua inglesa: Red, Green e Blue. Cada uma dessas três cores primárias tem um intervalo de valores de 0 até 255. Combinando os 256 possíveis valores de cada cor, o número total de cores fica aproximadamente 16,7 milhões (256 X 256 X 256). Assim a partir da combinação das intensidades dessas três cores é possível obter as demais cores. A representação gráfica do modelo RGB, é formada por um cubo, como mostra a Figura 2.4.

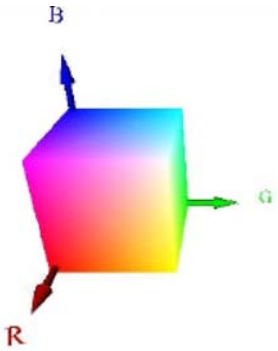


Figura 2.4: Representação gráfica do modelo RGB

Modelo HSB

O modelo de cor HSB se baseia na percepção visual humana das cores e não nos valores de RGB. Isso por que a mente humana não separa tão facilmente as cores em valores de vermelho/verde/azul ou ciano/magenta/amarelo/preto. O olho humano vê cores como

componentes de matiz, saturação e brilho. Em termos técnicos matiz se baseia no comprimento da onda de luz refletida de um objeto ou transmitida por ele. A saturação, também chamada de croma é a "quantidade de cinza" de uma cor. Quanto mais alta a saturação, mais intensa é a cor e brilho, que é a medida de intensidade da luz em uma cor. A Figura 2.5 mostra a representação gráfica do espaço de cores HSB.

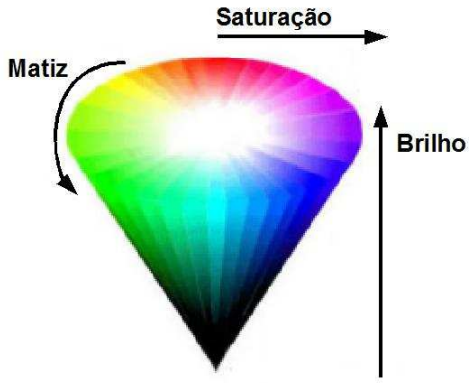
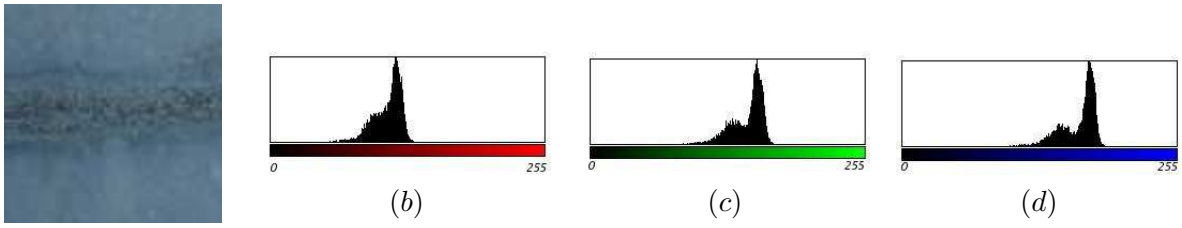


Figura 2.5: Representação gráfica do modelo HSB

Atributos de cores

Para cada um desses modelos de cores, utilizamos os valores da média da ocorrência dos píxels e a sua discretização com intervalos. Essas informações de R(red), G(green), B(blue), H(matiz), S(saturação) e B(brilho), foram capturadas das amostras e utilizadas como atributos, que ajudaram na criação dos conjuntos de treinamento para a classificação. Esses atributos foram capturados a partir da geração dos histogramas de cada observação. A Figura 2.6 ilustra dois exemplos de histograma, capturando seus valores RGB a partir de uma amostra de defeito do couro bovino.



(a) (b) (c) (d)

Figura 2.6: Exemplo da criação do histograma a partir de uma amostra de couro com defeito. (a) amostra com um risco de defeito, (b) histograma na componente R(Red) da amostra, (c) histograma na componente G(Green) da amostra, (d) histograma na componente B(Blue) da amostra.

2.3.4 Filtro de Gabor

A partir da técnica de extração de atributos baseada em funções de Gabor é possível representar de forma completa (freqüência e orientação) qualquer tipo de imagem. Esses conjuntos de funções são criados a partir de uma função base conhecida como Gabor principal [14].

As funções utilizadas nos filtros de Gabor são senóides complexas e bidimensionais modeladas por uma função Gaussiana também bidimensional. O objetivo principal dessas funções é extrair atributos para representar diferentes tipos de texturas presentes na imagem, que são descritas pela freqüência e orientação já definidas pelas funções senoidais [4].

Com filtros de Gabor é possível a manipulação de diversos parâmetros como freqüência, orientação, excentricidade e simetria. Através dessas várias combinações são formados os bancos de filtros de Gabor [14].

A implementação dos filtros de Gabor presente na biblioteca $SIGUS$ é baseada na seguinte família de funções de Gabor [14]

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}) \cos(2\pi \frac{x'}{\lambda} + \psi) \tag{2.3.1}$$

$$x' = x\cos\theta + y\sin\theta \qquad y' = -x\sin\theta + y\cos\theta \tag{2.3.2}$$

A partir dos parâmetros da função de Gabor é possível controlar as seguintes propriedades: λ determina o valor do comprimento de onda no núcleo, θ especifica o ângulo de inclinação das ondas paralelas do filtro, σ determina o desvio padrão da distribuição normal, ψ determina o tamanho da janela do núcleo e γ determina a excentricidade do núcleo. A equação 2.3.1 gera uma função senoidal modelada por uma função Gaussiana e a equação 2.3.2 rotaciona a equação 2.3.1 de acordo com o valor de θ.

Para a filtragem das imagens é utilizado o processo de convolução em duas dimensões da imagem I(x,y) com um núcleo de Gabor F(x,y). A imagem é convoluída com todo o banco de Gabor, onde se obtém uma resposta para cada núcleo. Assim são extraídas as características da imagem em cada um dos filtros[14].

2.4 Seleção e Redução de Atributos

No desenvolvimento de aplicações em visão computacional existe um enorme interesse na utilização da seleção ou redução de atributos, pois a alta dimensionalidade de atributos para esses tipos de aplicações pode provocar diversos problemas. Um desses problemas é quando temos uma quantidade pequena de amostras de treinamento e uma quantidade alta de atributos para sua representação. Dessa forma teremos maior chance de existir atributos menos discriminatórios, afetando assim o desempenho do classificador.

Como exemplo, dada uma imagem M, de largura w e altura h, se considerarmos cada píxel um atributo para a sua representação, sendo que T representa sua dimensão teremos então que $T = (w.h)$, em que o valor T para certas imagens pode ser muito alto. Podemos também considerar que qualquer alteração geométrica na imagem poderá também afetar o desempenho de classificação. Através dessas informações podemos verificar que a utilização de métodos de

redução de informações pode diminuir essa taxa de erro de forma robusta, diminuindo também seu tempo de aprendizagem.

A dimensionalidade do espaço de atributos pode resultar em dois tipos de problemas: alto custo de processamento e a geração do fenômeno conhecido como maldição da dimensionalidade. A Maldição da dimensionalidade pode ser caracterizada como uma degradação nos resultados de classificação, com o aumento da dimensionalidade dos dados [27]. Esse fenômeno ocorre quando o número de elementos de treinamento requeridos para que um classificador tenha um bom desempenho é uma função exponencial da dimensão do espaço de características. A partir disso veremos então duas soluções muito utilizadas em trabalhos correlatos, que tem o objetivo principal a minimização de informações para sua classificação, que são: seleção e redução de atributos.

Seleção de atributos é um problema de otimização de classificadores, com objetivo de encontrar a partir de um conjunto de atributos, um subconjunto com a melhor eficiência no processo de classificação[20]. O método de seleção de atributos pode ser visto como um processo de busca onde um algoritmo deve encontrar os melhores atributos dentro de um conjunto. A Figura 2.7 ilustra um breve exemplo da utilização da seleção de um subconjunto de atributos X_M a partir de um conjunto X_N, tal que, $X_M \leq X_N$.

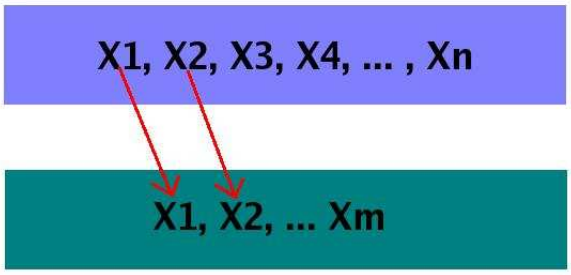
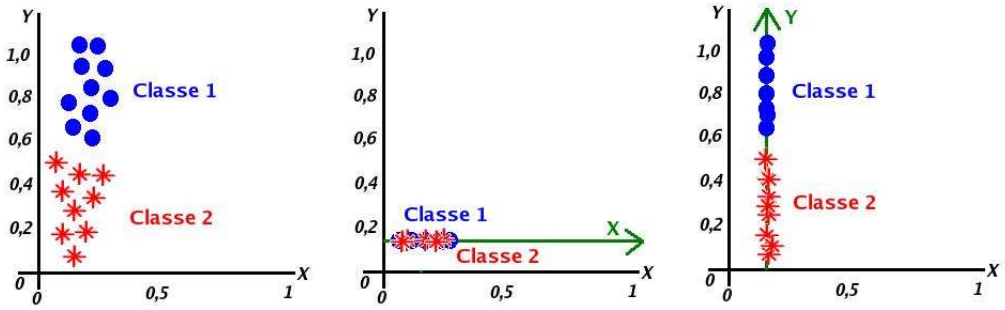


Figura 2.7: Exemplo da seleção de um sub-conjunto X_M, a partir de um conjunto de atributos X_N.

Para exemplificar melhor a idéia de seleção de atributos podemos analisar a Figura 2.8, em que temos a representação de um conjunto de dados de duas classes, sendo representado por dois atributos x e y. Se projetarmos os dados para cada atributo separadamente podemos verificar que o atributo y apresenta uma melhor discriminalidade entre as classes de classificação. Dessa forma podemos selecionar o atributo y, como sendo o melhor subconjunto, com a melhor taxa de discriminação entre os dados.

Similar à técnica de seleção de atributos a redução também é um problema de otimização, mas não tendo a característica de selecionar e sim de gerar novos atributos. Esse processo é realizado através de uma combinação entre um conjunto de atributos, criando assim um novo conjunto, que possa manter a mesma eficiência no processo de classificação[20]. A Figura 2.9, ilustra um exemplo utilizando a redução de atributos para a geração de um novo conjunto Y_M a partir do conjunto X_N.

Para entendermos melhor a redução de atributos podemos analisar a Figura 2.10, que ilustra de modo mais detalhado a utilização da redução de atributos a partir da representação gráfica de duas classes ($classe_1, classe_2$) com 2 atributos x e y. Analisando a dispersão das classes e dos atributos podemos realizar uma combinação estatística entre o conjunto de informações, gerando



(a) (b) (c)

Figura 2.8: Exemplo da utilização de seleção de atributos, utilizando duas classes e 2 atributos x e y. (a) conjunto de amostras, (b) projeção dos dados no atributo x e (c) projeção dos dados no atributo y

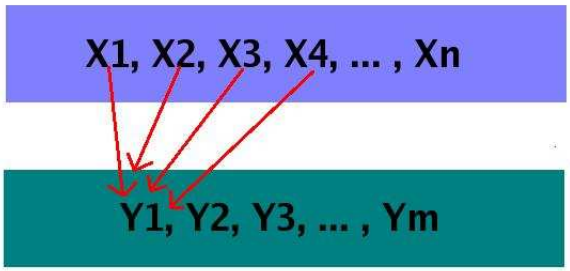


Figura 2.9: Exemplo da utilização da redução de atributos de um conjunto X_N, para a geração de um novo conjunto Y_M.

dessa forma um novo atributo z, que ao projetarmos os dados mantemos a discriminabilidade entre as classes.

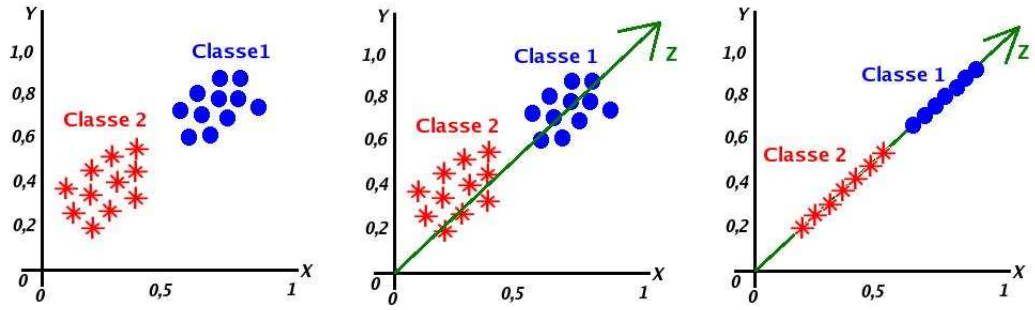


Figura 2.10: Exemplo da utilização da redução de atributos, utilizando duas classes e 2 atributos x e y. (a) conjunto de amostras, (b) geração do novo atributos e (c) projeção dos dados para o novo atributo gerado.

Antes de focarmos na técnica de Fisher iremos descrever inicialmente a técnica de Análise de Componentes Principais utilizada na redução de dimensão. O objetivo de abordar esse tópico nesse contexto é demonstrar a utilização de uma técnica de redução de informações em que sua análise se limita somente em verificar o conjunto de dados, sem levar em consideração a dispersão entre classes de classificação e assim apresentar a técnica de Fisher de maneira mais intuitiva que amplia esses conceitos.

2.4.1 Análise de Componentes Principais - PCA

A técnica de Análise de Componentes Principais tem o objetivo de procurar uma melhor forma de representar os dados a partir da redução do seu espaço de atributos, de tal forma que sua representação seja a mais representativa e reduzida.

O método de PCA transforma um vetor qualquer $Z_1 \epsilon \Re^n$ em outro vetor $Z_2 \epsilon \Re^m$ tal que, $m \leq n$, projetando as amostras nas m direções ortogonais de maior variância, chamados aqui de componentes principais. Estes componentes são responsáveis pela indicação da variância das observações, sendo representada de forma reduzida, descartando assim os atributos sem grande variância.

Para realizar o cálculo dos componentes principais basta utilizar informações da matriz de covariância do conjunto de amostras e em seguida calcular os autovalores e auto-vetores correspondentes, afim de se encontrar as direções que indicam a maior variância do conjunto de amostras. De forma mais simplificada podemos considerar que as componentes principais são os auto-vetores da matriz de covariância, em que o primeiro componente principal é um auto-vetor referente ao maior auto-valor e o segundo componente principal é um auto-vetor associado ao segundo maior auto-valor, e assim sucessivamente.

Considere um conjunto de amostras contendo n classes c, sendo representado por dois atributos x e y. Para acharmos o componente principal inicialmente calculamos a matriz de variância e covariância do conjunto de amostras. Em seguida calculamos a matriz de auto-vetores e auto-valores referente ao resultado da matriz de covariância e por fim capturamos

o vetor referente ao maior auto-valor, sendo a componente principal. A Figura 2.11, ilustra graficamente o comportamento final na projeção da componente principal, indicando a maior variância do conjunto de amostras.

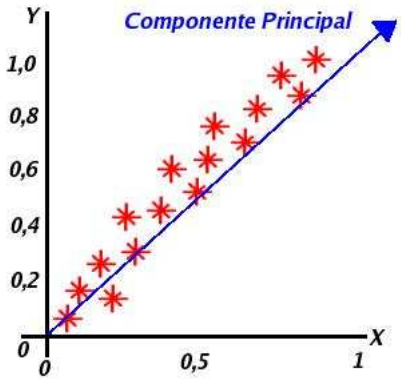


Figura 2.11: Exemplo da projeção de um conjunto de amostras sobre a componente principal.

2.5 Couro Bovino

O couro bovino é a pele curtida de animais, utilizada como material nobre em diversos segmentos industriais. Por ser um produto natural perecível necessita de cuidados. Isso ocorre devido à sua constituição química, formada pelos seguintes componentes: água(60%), proteínas(33%), gorduras(2%), sais minerais (0,5%) e outras substâncias (0,5%) [13].

A anatomia do couro bovino pode ser dividida em três partes que são: epiderme[1] (camada superior), a derme (camada intermediária, que resulta na pele comercializável) e a hipoderme[2] (camada inferior, encostada na carne do animal). A derme é a camada industrializada que se transformará em couro - produto estável e resistente, sendo a camada mais importante do couro bovino.

A classificação brasileira das regiões do couro bovino pode ser dividida em três partes: cabeça, grupon e flancos. Na classificação européia, a parte denominada cabeça é subdividida em paleta e colares e as demais partes são semelhantes. A parte mais nobre do couro bovino é o grupon (ou lombo ou dorso). Isso se deve por que suas fibras têm a melhor textura, maior uniformidade e resistência. A região da cabeça apresenta grande quantidade de rugas e pele de maior espessura. E por fim, os flancos que têm as fibras com textura inferior (vazias, abertas e finas) proporcionando mais facilidade em rupturas. A Figura 2.12 mostra essa divisão [11].

Dois tipos de estágios do pré-processamento do couro bovino serão descritos nesse trabalho: Couro Cru e Wet-Blue. Couro Cru é o couro em sua fase inicial, que ainda não passou por nenhuma fase de curtição. Wet-Blue é um couro após a fase de curtimento, mais especificamente na operação chamada rebaixe. A Figura 2.13 mostra esses processos.

Um dos grandes problemas em couros bovinos são seus diversos tipos de defeitos. Entre eles se destacam: berne (defeito encontrado similar a furos, causados pela larva da mosca conhecida

[1]Camada formada por queratina, em que serão eliminados no processo de depilação
[2]Camada formada por tecido adiposo, sendo eliminada na operação de descarne

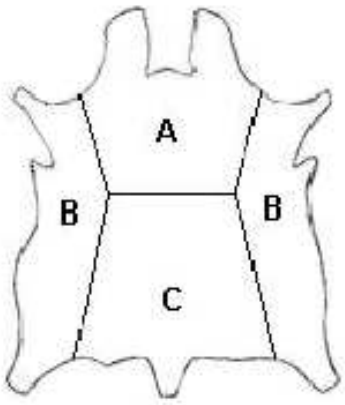
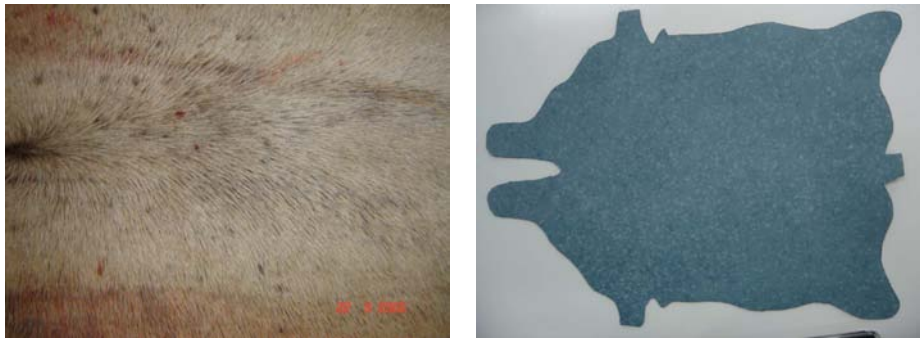


Figura 2.12: Partes do Couro Bovino (Classificação Brasileira). (A) Cabeça, (B) Flanco, (C) Grupão



(*a*) (*b*)

Figura 2.13: Imagens de couro bovino. (a) Imagem couro cru, (b) Imagem couro no estado Wet-Blue.

como "berne"), carrapato (marcas feitas pelo "carrapato"que aparecem nos couros com maior flor[3] Lixada), cortes de esfola (são cortes que aparecem no couro, causados por faca, quando os animais são abatidos), marca de fogo (defeitos causados por marcas de identificação do animal, causando grandes prejuízos nos couros), riscos (defeitos normalmente causados por chicote ou arame farpado) e veias (problemas arteriais do animal, em que problemas de estrutura ou rompimento se alargam e ficam perto da flor, aparecendo após o curtimento) [13]. A Figura 2.14 mostra alguns tipos de defeitos em Couro Cru e Wet-Blue.

A alta qualidade do couro é muito importante em diversos seguimentos da indústria como, sapatos, bolsas, roupas etc. A sua boa aparência em produtos fabricados usando couro depende de regiões nobres que não apresentam defeitos, ou seja, características na superfície do couro que possam prejudicar a aparência final do produto. O couro bovino, em particular apresenta defeitos que ocorrem desde a fase produtiva do animal até o seu abate.

[3]Parte externa do couro bovino, em que antes do uso é submetida a tratamentos especiais.

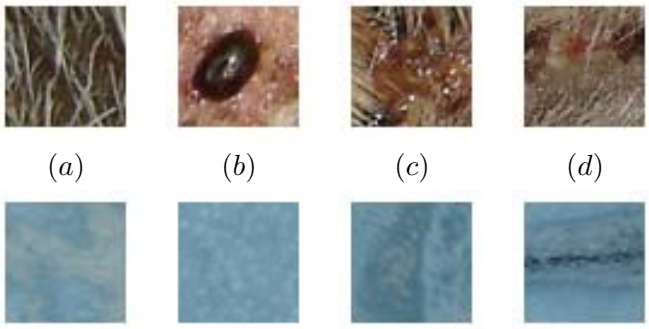


Figura 2.14: Imagem de defeitos de couros bovino nos processos (Couro Cru e Wet-Blue). (a) e (e) defeito sarna, (b) e (f) defeito carrapato, (c) e (g) defeito marca fogo, (d) e (h) defeito risco.

Para entendermos melhor o problema da qualidade do couro bovino iremos apresentar pesquisas realizadas exclusivamente na região de Mato Grosso do Sul, com base na cartilha da Embrapa [12], relatando sobre o problema crítico da produção do couro sobre a região Centro-Oeste. Os principais motivos que levaram a Embrapa a realizar essa análise sobre o estado de Mato Grosso do Sul foram as seguintes:

- É o principal couro produzido no país, tanto em qualidade quanto em demanda de mercado;
- Mato Grosso do Sul é o maior produtor de couro bovino, produzindo em média 4,5 milhões anuais, o que representa assim 14% da produção nacional;

A Tabela 2.1 identifica os principais defeitos que comprometem sensivelmente o couro bovino. As Tabelas 2.2 e 2.3 nos ajudam também a visualizar a classificação do couro brasileiro comparado a classificação do couro bovino nos EUA. E por fim podemos visualizar nas Tabelas 2.4, 2.5 e 2.6, uma breve comparação em termos de receita entre o Brasil e os EUA, para entendermos melhor o quanto o Brasil deixa de ganhar por causa da qualidade do couro bovino. Essas tabelas são apenas projeções do que poderia ser, se o País produzisse 85% de couro de primeira qualidade e exportasse toda sua produção.

Tabela 2.1: Principais defeitos que comprometem a qualidade do couro bovino no Brasil

Localização e Causa	%
Dentro da propriedade	60
- Causados por ectoparasitas (bernes, bicheiras, moscas, etc.)	40
- Manejo inadequado (marca a ferro, ferrão, etc.)	10
- Arame farpado, galhos e espinhos	10
Fora da propriedade	10
- Uso de guizo pontiagudo ou roseta para condução do gado	4
- Carrocerias com travessas quebradas/madeiras lascadas, pontas de prego, etc.	6
Nos Frigoríficos	25
- Esfola mal feita durante o abate	10
- Má conservação do couro	15

Tabela 2.2: Classificação do Couro Bovino no Brasil

Tipo	**Classificação**	**Percentual**	**Produção Anual**
AAA	1º	8%	2.560,00
AA	2º	22%	7.040,00
A	3º	35%	11.200,00
B	4º	25%	8.000,00
C	5º	7%	2.240,00
D	6º	3%	960,00
Total	-	-	32.500,00

Tabela 2.3: Classificação do Couro Bovino nos EUA

Tipo	**Classificação**	**Percentual**	**Produção Anual**
AAA	1º	85%	30.600,00
AA	2º	10%	3.600,00
A	3º	-	-
B	4º	-	-
C	5º	-	-
D	6º	5%	1.800,00
Total	-	-	36.000,00

Tabela 2.4: Comparação entre receitas entre Brasil e os Estados Unidos - Wet blue (preço mercado internacional US$ 50.00)

	Produção anual	**Produção AAA**	**Valor em US$**
Estados Unidos	36.000	30.600	1.530.000.000
Brasil	32.000	2.560	128.000.000
Perda de receita	-	-	1.253.250.000

Tabela 2.5: Comparação entre receitas entre Brasil e os Estados Unidos - Semi Acabado (preço mercado internacional US$ 60.00)

	Produção anual	**Produção AAA**	**Valor em US$**
Estados Unidos	36.000	30.600	1.836.000.000
Brasil	32.000	2.560	153.600.000
Perda de receita	-	-	1.503.900.000

Tabela 2.6: Comparação entre receitas entre Brasil e os Estados Unidos - Acabado (preço mercado internacional US$ 80.00)

	Produção anual	**Produção AAA**	**Valor em US$**
Estados Unidos	36.000	30.600	2.448.000.000
Brasil	32.000	2.560	204.800.000
Perda de receita	-	-	2.005.200.000

Capítulo 3

Trabalhos Correlatos

Nesta seção serão apresentados alguns trabalhos que têm como objetivo o estudo e aplicação de Análise Discriminante de Fisher em reconhecimento de padrões para detecção de defeitos. Para uma melhor análise essa seção será dividida em duas partes: trabalhos utilizando Análise Discriminante de Fisher e trabalhos sobre detecção de defeitos.

3.1 Análise Discriminante

No trabalho [34] é realizado um estudo aprofundado do problema da técnica de Fisher para a redução de um número alto de atributos e a criação de um novo método de redução baseada na técnica de análise discriminante. Essa nova técnica utiliza a aproximação direta da estabilização da matriz de dispersão intra-classe. Os experimentos foram realizados através do reconhecimento de faces, sendo comparados com outras técnicas como: Chen et al.'s Method (CLDA), Yu and Yang's Method (DLDA), Yang and Yang's Method (YLDA) e o novo método proposto (NLDA). Como resultado foi verificado que o método proposto melhora o desempenho da classificação baseada na análise discriminante quando a matriz de dispersão intra-classe é ou não singular.

Em [1], utiliza-se também análise discriminante para resolver problemas de seleção de atributos, afim de melhorar o desempenho em termos de classificação. São demonstradas também técnicas que foram utilizadas para análise dos resultados utilizando métodos discriminantes como: os métodos de seleção Teste F, Teste Bootstrap e Métodos robustos. Como resultado, esse trabalho mostra que a metodologia bootstrap pode ser uma ferramenta importante para análise de múltiplos problemas que surgem no dia a dia, uma vez que permite "estimar" a distribuição de estimadores sem qualquer conhecimento sobre a distribuição da população ou populações.

No trabalho [5], apresenta-se estudos realizados de métodos de extração de características e de classificação estatística para a aplicação em reconhecimento de faces. Mais especificamente, é apresentada a técnica para redução de dimensionabilidade análise discriminante. Esse trabalho obteve como resultado utilizando métodos de análise estatística, que é possível obter-se um grande aumento no poder de reconhecimento de padrões e uma maior velocidade em seu reconhecimento.

Em [28] é realizado um estudo sobre o modelo de composição de especialistas locais (CEL) e

um estudo sobre análise discriminante para classificação de dados. Nesse trabalho, experimentos com base em dados reais foram utilizados, para medição de desempenho e comparação entre o modelo CEL e análise discriminante. Para a realização dos testes utilizando análise discriminante foram utilizadas três técnicas: Análise Discriminante de Fisher, Regressão Logística e Extended DEA-DA. Os resultados desse trabalho mostram que o modelo de classificação CEL e análise discriminante, obtiveram um grande êxito na classificação dos seus dados, que por uma pequena diferença na taxa de classificação correta a análise discriminante obteve o maior sucesso chegando a 91,6% de acerto.

Em [39] é criada uma proposta para solucionar o problema de redução de atributos utilizando algoritmo de análise discriminante, com aplicações em reconhecimento de faces. A sugestão de mudança é criar um algoritmo para a eliminação do espaço nulo da matriz de dispersão de inter-classe, o qual o trabalho defende que não há informação útil. O resultado desse trabalho é um algoritmo unificado de análise discriminante, dando uma exata solução ao problema do critério de Fisher quando ocorre uma singularidade na matriz de dispersão intra-classe.

No trabalho [9] é apresentado um novo método de extração de atributos em reconhecimento de padrões, baseado na interpretação geométrica da Análise Discriminante de Fisher. A nova técnica proposta, chamada neste trabalho de: "simple-FLDA" é executada apenas como um simples algoritmo interativo, não utilizando assim, qualquer computação de matrizes. Essa nova técnica é baseada na interpretação geométrica na maximização da variância da matriz inter-classes e minimização da variância da matriz intra-classes. Para análise do desempenho da nova técnica criada foi utilizado como aplicação o reconhecimento de moedas e faces. Como resultado foi verificado que para as duas aplicações, o método "simple-FLDA" apresentou melhores resultados de classificação do que a técnica de Análise de Componentes Principais.

No trabalho [38] é apresentada detalhadamente a comparação de duas soluções para o problema de singularidade, utilizando a técnica de Fisher. As técnicas apresentadas são: (NLDA) que resolve o problema de singularidade, maximizando a matriz inter-classe, através do espaço nulo da matriz de dispersão intra-classe e a técnica (OLDA), que resulta na transformação ortogonal do espaço nulo da matriz de dispersão intra-classe. Como conclusão desse trabalho foi verificada que utilizando as duas técnicas em aplicações como: reconhecimento de textos em documentos e reconhecimento de faces, as mesmas apresentaram o mesmo desempenho na redução de atributos.

Em [17] foram apresentadas duas novas técnicas de redução de atributos, que tem como objetivo solucionar o problema de SSSP (*Small Sample Size Problem*). Os métodos apresentados como propostas são as técnicas NLDA, técnica baseado na análise do espaço nulo da matriz de dispersão intra-classe e o método NKLDA, que também utiliza os mesmos conceitos integrados com a técnica baseada em Kernel. São apresentados também trabalhos sobre as técnicas LDA e DLDA. Para medir o desempenho das técnicas propostas, foram realizados experimentos sobre os bancos de imagens de faces ORL e FERET. Como resultado foi verificado que a técnica baseada em Kernel NKLDA é melhor do que NLDA, para redução de um número alto e baixo de atributos.

No trabalho [36] é apresentado um novo algoritmo que unifica conceitos relacionados às técnica LDA/PCA para reconhecimento de faces. Este novo algoritmo maximiza diretamente o critério LDA sem o passo de separação utilizando PCA mostrando que essa maximização é possível sem perda de informações na sua utilização. Para mostrar o desempenho da nova técnica citada, foram realizados experimentos utilizando o banco de imagens "Olivetti-Oracle", conhecido também como ORL, que consiste de 400 imagens de faces frontais. Como melhor

resultado de redução, foi obtida uma taxa de 95% de acerto, sem que houvesse qualquer pré-processamentos nas imagens das faces.

3.2 Detecção de Defeitos

O objetivo dessa seção é apresentar alguns trabalhos sobre detecção de defeitos, que serão utilizados como base para a realização dos experimentos utilizando redução de atributos. Entre os trabalhos pesquisados se destaca [31], que apresenta uma nova metodologia para detecção de defeitos em couro bovino, baseado na transformada Wavelets. A metodologia criada utiliza um banco de filtros otimizados, onde cada filtro é ajustado a um tipo de defeito. Esses tipos de filtros e as faixas do wavelet são selecionados baseados na maximização dos atributos capturados dos defeitos e regiões do couro. Esse tipo de metodologia pode detectar defeitos mesmo quando são apresentadas pequenas variações nos atributos, as quais, nas técnicas mais genéricas de detecção de defeitos em textura, não são encontradas. Além disso, a mesma se comporta de maneira rápida na detecção de defeitos em tempo real.

Em [16] é apresentado um novo método para detecção de defeitos baseado em histograma. Esse trabalho mostra a utilização do critério χ^2 ($chi - quadrado$), como é ilustrado na fórmula 3.2.1, para análise de imagens, sendo um dos critérios para a construção do histograma.

$$\chi^2 = \sum_i \frac{(R_i - S_i)}{(R_i + S_i)} \tag{3.2.1}$$

As variáveis R_i e S_i são respectivamente a contagem dos pixels do nível de cinza dos histogramas e de outra área da imagem e a variável i é o número de pixels referente a área da imagem. A técnica apresentada consegue detectar as áreas defeituosas do couro, baseando-se no cálculo da diferença entre o histograma em nível de cinza com as outras áreas procuradas na imagem.

No Trabalho [21] é apresentada uma modificação na técnica de extração de atributos chamada de (Local Binary Pattern) para a detecção de defeitos em textura. Este trabalho utiliza a técnica em duas fases, que são treinamento e classificação. Na fase de treinamento o método LBP é aplicado a todas as linhas e colunas, pixel a pixel das amostras de defeitos, que deverão ser identificados criando assim um vetor de atributos. Em seguida na fase de classificação, a imagem a ser utilizada na detecção é dividida em janelas, sendo cada uma delas binarizadas e sendo comparadas com o vetor de atributos. A partir disso foi possível verificar que com as modificações realizadas na técnica LBP é possível se obter uma taxa de acerto na detecção de defeitos superior a 95%.

Capítulo 4

Análise Discriminante de Fisher

A Análise Discriminante de Fisher (FLDA), é uma técnica que se tornou muito comum em aplicações de visão computacional. Essa técnica utiliza informações das classes associadas a cada padrão para extrair linearmente os atributos mais discriminantes. Através da Análise Discriminante de Fisher podemos realizar a discriminação entre classes, através de processos supervisionados (quando se conhece sua classe de classificação) ou através de processos não supervisionados, a qual a classe referente à amostra não é conhecida.

Na técnica de FLDA, quando utilizada como método supervisionado, é importante que algumas condições sejam atendidas, como: (1) as classes sob investigação devem ser mutuamente exclusivas, (2) cada classe deve ser obtida de uma população normal multi-variada, (3) duas medidas não podem ser perfeitamente correlacionadas, entre outras [34].

4.1 Redução de Atributos utilizando Fisher

Nos trabalhos [3], [39] e [41] é apresentada detalhadamente a utilização da Análise Discriminante de Fisher. Essa técnica consiste na computação de uma combinação linear de m variáveis quantitativas que mais eficientemente separam grupos de amostras em um espaço m-dimensional fazendo com que a dispersão intra-classes e inter-classes seja maximizada.

A separação intra-classe e inter-classe são descritas através das fórmulas 4.1.1 e 4.1.2, estabelecidas por Fisher.

1. Dispersão intra-classes:

$$S_w = \sum_{j=1}^{n_c} \sum_{i=1}^{T_j} (x_i^j - u_j).(x_i^j - u_j)^t, \tag{4.1.1}$$

   em que x_i^j é a i-ésima amostra da classe j, u_j é a média da classe j, T_j é o número de amostras da classe j e n_c é o número de classes;

2. Dispersão inter-classes:

$$S_b = \sum_{j=1}^{n_c} (u_j - u).(u_j - u)^t, \tag{4.1.2}$$

   em que u é a média de todas as classes, ou seja,

$$u_j = \frac{1}{T_c} \sum_{j \epsilon c} x_j, \tag{4.1.3}$$

$$u = \frac{1}{T} \sum_{j}^{c} T_c u_j, \tag{4.1.4}$$

e T_c, é o número de amostras da classe c.

A partir do cálculo de dispersão intra-classe e inter-classe de um conjunto de amostras, é possível seguir o critério de Fisher, maximizando a medida inter-classes e minimizando a medida intra-classes. Uma forma de fazer isso é maximizar a taxa $S_f = S_w^{-1}.S_b$.

Em Análise Discriminante a redução de atributos é realizada a partir de um conjunto de amostras para n_c classes, tendo p atributos, com o objetivo de reduzir para m atributos. Para a redução de atributos por Fisher segue-se o seguinte procedimento, ilustrado pelo Algoritmo 1.

1. Calcular a dispersão S_w e S_b para n_c classes;
2. Maximizar a medida inter-classes e minimizar a medida intra-classes S_f a partir de $S_w^{-1}.S_b$.

Algoritmo 1: Algoritmo FLDA

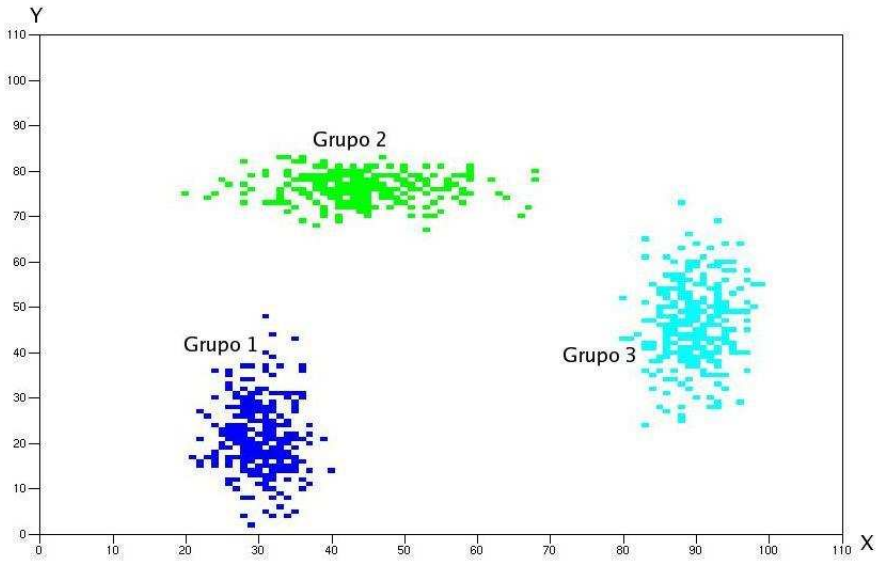
Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.
Saída: Matriz Reduzida $FLDA$

```
1 Sw ⟵ calculaConvarianciaIntraClasse(amostras, M);
2 Sb ⟵ calculaCovarianciaInterClasse(amostras, M);
3 Sf ⟵ Sw.inversa.multiplica(Sb);
4 Ev ⟵ Sf.analisaAutoValoresAutoVetores();
5 autoVetores ⟵ Ev.autoVetores();
6 autoValores ⟵ Ev.autoValores();
7 FLDA ⟵
  buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R);
8 retorna matrizReduzida(amostras,FLDA)
```

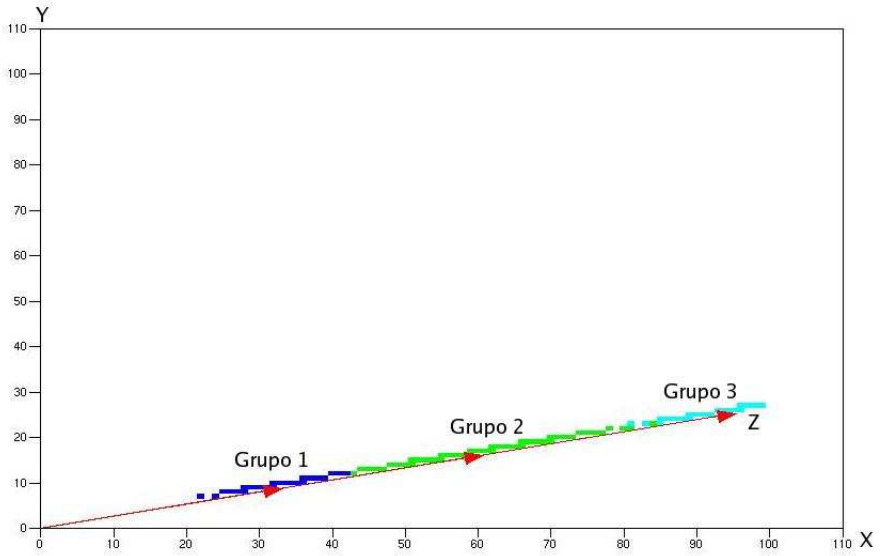
A partir de S_f é possível a redução de atributos com base em seus auto-valores e auto-vetores, em que os m atributos selecionados correpondem aos m maiores auto-valores com maior grau de seus auto-vetores. A Figura 4.1(a) ilustra 3 conjuntos de amostras representando cada classe de classificação e utilizando dois atributos x e y e a Figura 4.1(b) mostra um exemplo usando FLDA para a redução de 2 atributos x e y para um novo atributo z.

4.2 FLDA Vs. PCA

Em diversos trabalhos, a técnica de Análise Discriminante de Fisher é citada junto ao trabalho de Análise de Componentes Principais. Isso se deve ao fato de que PCA possui também um bom desempenho na redução de atributos, mas com características diferentes à técnica de



(a)



(b)

Figura 4.1: Exemplo utilizando Análise Discriminante de Fisher para redução de atributos. (a) conjunto de amostras utilizando 2 atributos x e y, (b) redução de atributos para um único atributo z

FLDA. Nessa seção iremos descrever brevemente as principais diferenças, quanto a seus conceitos e aplicações da utilização da técnica de Fisher e PCA.

No trabalho [3], os autores comparam a técnica de PCA com a técnica de FLDA, e experimentalmente mostram que o espaço de atributos reduzido pela técnica de FLDA, apresentou melhores resultados de classificação do que o espaço reduzido por PCA em aplicações de reconhecimento de faces. A técnica de PCA também é um método de redução de atributos, mas diferente da técnica de FLDA, sua análise não considera a distribuição das classes de classificação. Com isso, alguns problemas podem ser encontrados na utilização do PCA, como é mostrado na Figura 4.2, ilustrando o vetor reduzido PCA junto com sua projeção.

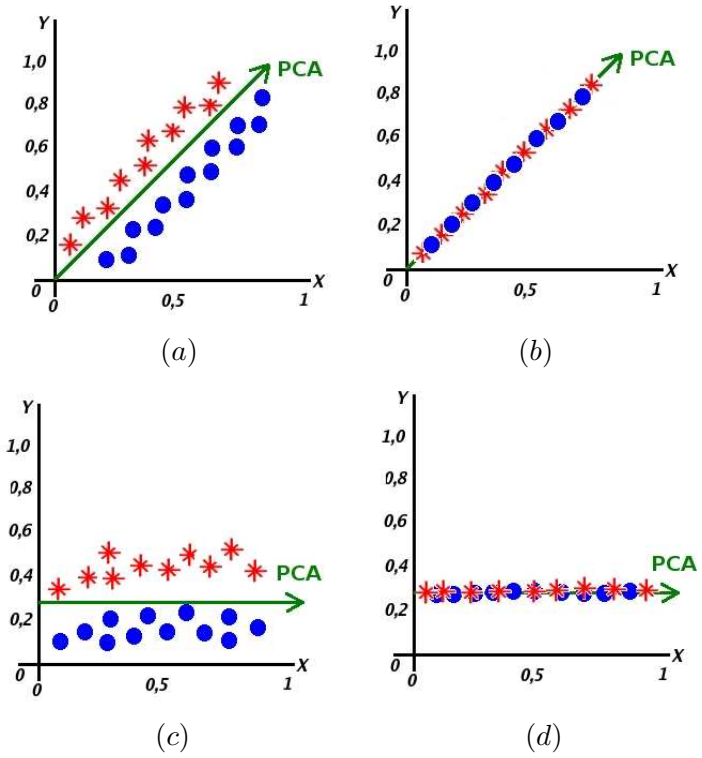


Figura 4.2: Exemplo da técnica de PCA aplicada na redução de atributos, (a) e (c) conjunto de amostras, (b) e (d) suas respectivas projeções.

Utilizando a técnica de FLDA esses problemas podem ser resolvidos. A Análise Discriminante de Fisher é uma técnica que além de utilizar informações associadas ao espaço de amostras, leva também em consideração a distribuição das classes no espaço original. Na Figura 4.3, podemos visualizar um exemplo da redução de dimensão utilizado FLDA e PCA, em que a técnica de FLDA apresenta melhores resultados na classificação.

4.3 Problemas Encontrados

Um problema crítico na utilização da Análise Discriminante de Fisher está na singularidade e instabilidade da matriz de dispersão intra-classes. Em algumas aplicações, como reconhecimento de padrões em textura, onde existe um número alto de píxels, e o número de atributos é superior à

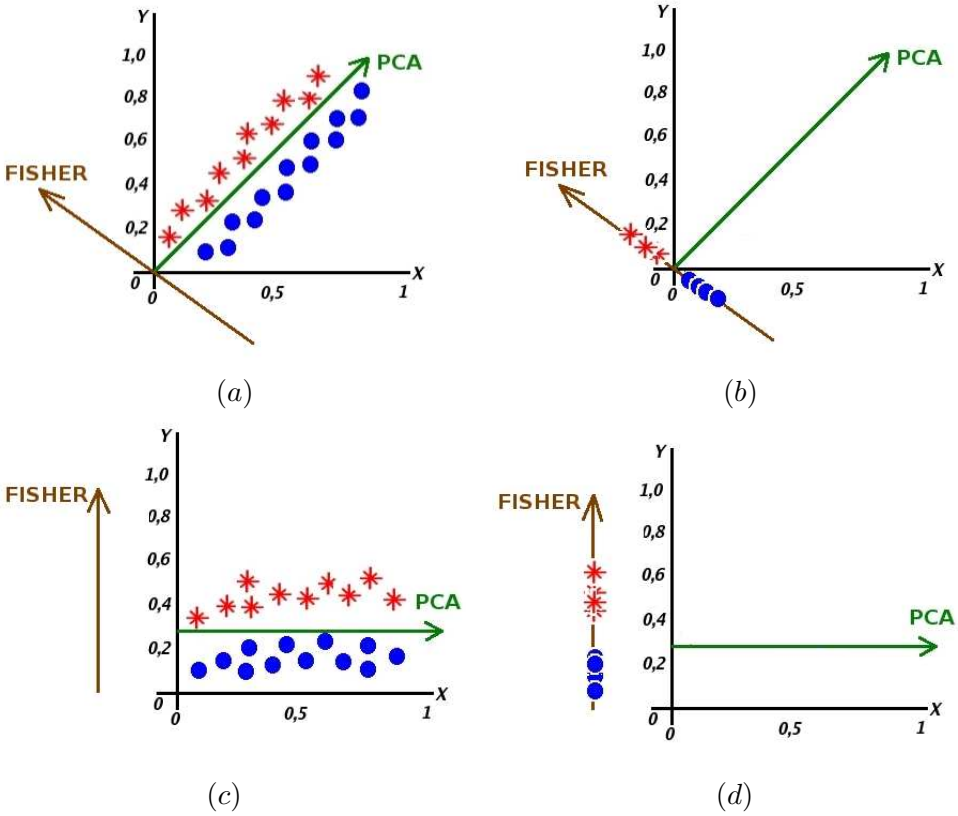


Figura 4.3: Exemplo da Técnica de Fisher aplicada na redução de atributos, (a) e (c) conjunto de amostras, (b) e (d) suas respectivas projeções.

quantidade de amostras de treinamento, a utilização da técnica FLDA fica limitada. Isto implica que a matriz de dispersão intra-classes será singular, caso seu *posto* seja menor que o número de amostras ou se o número total de amostras de treinamento não é significativamente maior que a dimensão do espaço de características.

O foco principal da técnica de FLDA está sobre o critério estabelecido por Fisher 4.3.1. Caso a matriz intra-classes seja não singular, o problema de maximização pode ser transformado em um problema de auto-vetores e auto-valores. Atualmente em diversas aplicações, existem situações onde o número de atributos é muito maior do que a quantidade de amostras, resultando assim no problema ($p >> n$), em que p é o número de atributos e n número de amostras e dessa forma a matriz intra-classes será singular.

$$arg_c max \frac{|c^T S_b c|}{|c^T S_w c|}, \tag{4.3.1}$$

Algumas soluções para o redimensionamento de atributos utilizando FLDA consistem em modificar a matriz intra-classes através de pequenas perturbações diagonais utilizando técnicas como SVD[1] ou realizar uma análise detalhada sobre seu espaço nulo. Nesse trabalho, iremos analisar e implementar essas técnicas com objetivo de comparar e verificar seu desempenho a partir de aplicações sobre problemas reais.

4.4 Soluções Lineares

4.4.1 FisherFaces

Fisherfaces é um método de redução de atributos que foi criado a partir do problema da utilização da técnica tradicional de LDA. Esse problema é muito encontrado em aplicações com um número alto de atributos, como reconhecimento de faces onde o *posto* da matriz de dispersão S_w da técnica de $FLDA$ não é maior do que $(N-c)$, sendo c a quantidade de classes, e geralmente o número de imagens para o treinamento N é muito menor do que o número de atributos de cada imagem de amostra [3].

A técnica de FisherFace possui uma grande vantagem sobre outras técnicas pelo fato de se usar informações das classes para a remoção de informações não discriminatória e identificando as projeção que mais separam suas classes.

Para isso FisherFace propõe uma alternativa no critério de Fisher, que está em encontrar uma projeção que reduza o espaço de características utilizando Análise de Componentes Principais (PCA) e em seguida Análise Discriminante Linear, para achar as características mais discriminante sobre o sub-espaço PCA reduzido.

De forma detalhada podemos calcular a projeção da matriz $P_{FisherFaces}$ da seguinte forma:

1. Cálculo da projeção de FisherFaces;

$$P_{FisherFaces} = P_{lda} * P_{pca}, \tag{4.4.1}$$

[1]Decomposição de valor singular é uma técnica matemática muito utilizada em alguns casos que necessitam achar a inversa uma matriz, com problemas de singularidade.

em que P_{pca} é a matriz de projeção do espaço original da imagem referente ao sub-espaço de PCA, em que é reduzido para um espaço de atributos de dimensão (N-c) e P_{lda} é a matriz de projeção do sub-espaço PCA e sub-espaço LDA extraído, sendo reduzido para c-1 atributos, como mostra a fórmula 4.4.3.

2. Cálculo da projeção de P_{pca};

$$P_{pca} = argmax \mid P^T S_T P \mid \tag{4.4.2}$$

3. Cálculo da projeção de P_{lda};

$$P_{lda} = argmax \frac{\mid P^T P_{pca}^T S_b P_{pca} P \mid}{\mid P^T P_{pca}^T S_w P_{pca} P \mid} \tag{4.4.3}$$

A partir disso podemos verificar que se $P_{pca}^T S_w P_{pca}$ é não singular, então o critério de Fisher é maximizado, quando a matriz de projeção P_{lda} é composta dos auto-vetores de $(P_{pca}^T S_w P_{pca})^{-1}(P_{pca}^T S_b P_{pca})$ contendo $(c-1)$ auto-valores.

Algoritmo

Para exemplificarmos melhor a idéia das técnicas aqui apresentadas como soluções para o problema de singularidade, iremos apresentar para cada técnica o algoritmo utilizado, que auxiliaram na implementação das técnicas tanto na linguagem Scilab quanto na linguagem Java. O algoritmo 2 de FisherFace é ilustrado abaixo.

Algoritmo 2: Algoritmo FisherFace

Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.
Saída: Matriz Reduzida $FisherFace$

```
1  Sw ⟵ calculaConvarianciaIntraClasse(amostras, M);
2  Sb ⟵ calculaCovarianciaInterClasse(amostras, M);
3  Ev ⟵ Sw.analisaAutoValoresAutoVetores();
4  autoVetores ⟵ Ev.autoVetores();
5  autoValores ⟵ Ev.autoValores();
6  PCA ⟵ buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R);
7  PCATransposta ⟵ PCA.transposta();
8  Sf ⟵ PCATransposta.multiplica(Sw).multiplica(PCA);
9  SfInversa ⟵ Sf.inversa();
10 Sf ⟵ SfInversa.multiplica(PCATransposta.multiplica(Sb).multiplica(PCA));
11 Ev ⟵ Sf.analisaAutoValoresAutoVetores();
12 autoVetores ⟵ Ev.autoVetores();
13 autoValores ⟵ Ev.autoValores();
14 FISHERFACE ⟵
   buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R);
15 retorna matrizReduzida(amostras,FISHERFACE)
```

4.4.2 Chen et al.'s Method (CLDA)

A técnica CLDA é uma proposta baseada no método LDA, com objetivo de solucionar o problema de singularidade relacionado ao uso da técnica de LDA em casos de aplicações com um número pequeno de amostras. O objetivo de CLDA é usar qualquer informação discriminante do espaço nulo da matriz de dispersão intra-classes, tentando aumentar ao máximo a matriz de dispersão inter-classes sempre que a matriz intra-classes seja singular.

Para isso CLDA propõe um método mais claro que faz o uso do espaço nulo da matriz de dispersão intra-classe S_w. A idéia básica desse método está em projetar todas as amostras sobre o espaço nulo de S_w, em que o resultado da matriz de dispersão será zero e assim será maximizada a matriz de dispersão inter-classes S_b [17].

A proposta é utilizar os auto-vetores correspondentes aos maiores auto-valores da matriz $(S_b + S_w)^{-1}S_b$, sempre que S_w é não-singular. Resumidamente encontrar a solução dos auto-vetores de $(S_b + S_w)^{-1}$ é similar ao critério de Fisher $S_w^{-1}.S_b$.

De modo detalhado podemos realizar o cálculo de CLDA P_{clda}, obedecendo os seguintes passos abaixo, e para facilitar seu entendimento segue o Algoritmo 3.

1. Inicialmente é calculado o *posto* da matriz de dispersão intra-classes S_w;
2. em seguida verificamos se S_w é não-singular. Se $posto = n$, então P_{clda} é representado como os auto-vetores correspondentes aos maiores auto-valores de $(S_b + S_w)^{-1}S_b$;
3. caso seja singular, calcula-se a matriz de auto-vetores $V = [V_1, ..., V_r, V_{r+1}, ..., V_n]$ da matriz de dispersão intra-classes S_w;
4. a matriz de dispersão P_{clda} é composta dos auto-vetores, correspondentes aos maiores auto-valores de $QQ^T S_b(QQ^T)^T$, sendo Q a matriz contendo o espaço nulo de S_w, onde $Q = [V_{r+1}, V_{r+2}, ..., V_n]$ é uma sub-matriz de V. Com isso a técnica demonstra que os valores obtidos pela transformação de QQ^T são os vetores mais discriminantes do espaço de amostra original.

4.4.3 Yu and Yang's Method (DLDA)

Yu e Yang [39] desenvolveram um método para tratar o problema de grande dimensão de dados em aplicações como reconhecimento da faces, buscando realizar a diagonalização simultânea das matrizes simétricas inter-classes (S_b) e intra-classes (S_w), chamado aqui de (DLDA). O objetivo do algoritmo de DLDA é remover inicialmente o espaço nulo de inter-classes e em seguida analisar a matriz de dispersão intra-classes.

Este método propõe um novo algoritmo que incorpora o conceito de espaço nulo. Primeiramente é removido o espaço nulo da matriz de dispersão intra-classe S_b e então é procurada uma projeção para minimizar a matriz de dispersão intra-classes S_w, chamado aqui de DLDA. O objetivo desse cálculo é por que o posto de S_b é menor do que de S_w. Isso significa que ao remover o espaço nulo de S_b, poderá assim remover inteiramento ou parte do espaço nulo de S_w, o qual é muito provável ser $posto - completo$ depois da operação de remoção [17].

DLDA é uma solução criada de modo inverso a técnica tradicional de LDA, a qual descarta o espaço nulo de intra-classes, que contém informações discriminantes. Esse processo de análise

Algoritmo 3: Algoritmo CLDA

Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.

Saída: Matriz Reduzida $CLDA$

```
1  Sw ⟵ calculaCovarianciaIntraClasse(amostras, M);
2  Sb ⟵ calculaCovarianciaInterClasse(amostras, M);
3  posto ⟵ Sw.posto();
4  se posto = M então
5      Sf ⟵ Sb.soma(Sw).inversa().multiplica(Sb);
6      Ev ⟵ Sw.analisaAutoValoresAutoVetores();
7      autoVetores ⟵ Ev.autoVetores();
8      autoValores ⟵ Ev.autoValores();
9      CLDA ⟵
       buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R);
10 senão
11     Ev ⟵ Sw.analisaAutoValoresAutoVetores();
12     autoVetores ⟵ Ev.autoVetores();
13     autoValores ⟵ Ev.autoValores();
14     Q ⟵ autoVetores.capturaSubMatriz(0, autoVetores.numeroLinhas() −
       1, posto − 1, autoVetores.numeroColunas() − 1);
15     QTRANS ⟵ Q.multiplica(Q.transposta());
16     PCLDA ⟵ QTRANS.multiplica(Sb);
17     PCLDA ⟵ PCLDA.multiplica(QTRANS.transposta());
18     Ev ⟵ PCLDA.analisaAutoValoresAutoVetores();
19     autoVetores ⟵ Ev.autoVetores();
20     autoValores ⟵ Ev.autoValores();
21     CLDA ⟵
       buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R);
22 retorna matrizReduzida(amostras,CLDA)
```

também evita problemas com singularidade relacionados ao uso da técnica tradicional de LDA, em grandes dimensões de dados, em que intra-classes tende a ser singular.

De modo detalhado podemos realizar o cálculo de CLDA P_{clda}, obedecendo os seguintes passos abaixo, e para facilitar seu entendimento segue o Algoritmo 4.

1. Inicialmente deve-se diagonalizar S_b, a partir do calculado da matriz de auto-vetores V, tal que $V^T S_b V = \Lambda$;
2. Seja Y as primeiras m colunas de V correspondente aos maiores auto-valores de S_b, em que $m \leq posto(S_b)$, calcula-se então $D_b = Y^T S_b Y$, em que D_b é a diagonal da sub-matriz mxm da matriz de auto-valores de Λ;
3. Considerando $Z = Y D_b^{-1/2}$ sendo uma transformação de S_b, podemos realizar a redução de dimensão de m para n atributos, isto é, $Z^T S_b Z = (Y D^{-1/2})^T S_b (Y D^{-1/2}) = I$;
4. Através da diagonalização de $Z^T S_w Z$, é calculado U e D_w, tal que, $U^T (Z^T S_w Z) U = D_w$;
5. Calcula-se então a matriz de projeção P_{dlda} através de $P_{dlda} = D_w^{-1/2} U^T Z^T$.

Algoritmo 4: Algoritmo DLDA

Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.
Saída: Matriz Reduzida $DLDA$

```
1  Sw ⟵ calculaCovarianciaIntraClasse(amostras, M);
2  Sb ⟵ calculaCovarianciaInterClasse(amostras, M);
3  Sf ⟵ Sb.multiplica(Sw.inversa());
4  Ev ⟵ Sb.analisaAutoValoresAutoVetores();
5  V ⟵ Ev.autoVetores();
6  A ⟵ Ev.autoValores();
7  Y ⟵ buscaAutoVetoresMaiorGrauAutoValores(V, A, R);
8  Db ⟵ Y.transposta().multiplica(Sb);
9  Db ⟵ Db.multiplica(Y);
10 Z ⟵ Y.multiplica(Db.potencia(−0.5));
11 evv ⟵ Z.transposta().multiplica(Sw).multiplica(Z);
12 Ev ⟵ evv.analisaAutoValoresAutoVetores();
13 U ⟵ Ev.autoVetores();
14 Dw ⟵ Ev.autoValores();
15 A ⟵ U.transposta().multiplica(Z.transposta());
16 DLDA ⟵ A.transposta().multiplica(Sw.potencia(−0.5));
17 retorna matrizReduzida(amostras,DLDA)
```

4.4.4 Yang and Yang's Method (YLDA)

Yang e Yang [35], recentemente criaram um novo método de extração de características, chamado aqui de YLDA. Esse método é capaz de derivar informações discriminatórias, utilizando somente os critérios de LDA para casos de matrizes singulares. Similar ao método de FisherFaces,

que é descrito na seção (4.3.1), o YLDA pode ser considerado também uma técnica de redução de dimensão, dividida em duas etapas. Para a realização da primeira etapa é utilizada a técnica de PCA, com objetivo de reduzir a dimensão do espaço original. Em seguida na segunda etapa é utilizado um algoritmo particular baseado em Fisher, chamado aqui de Optimal Fisher Linear Discriminant (OFLD), que será explicado em seguida para encontrar as características mais discriminantes sobre o sub-espaço PCA.

De forma detalhada podemos descrever o método OFLD, através dos seguintes passos, e o Algoritmo 5 ilustra a técnica YLDA.

1. Sobre o espaço de m-dimensões da transformada de PCA, são calculadas as matrizes de dispersão intra-classes e inter-classes;
2. Calcula-se então a matriz de auto-vetores $V = v_1, v_2, ..., v_m$ da matriz intra-classes S_w. O primeiro auto-vetor q da matriz S_w, corresponde a um auto-valor não zero;
3. A matriz de projeção $P_1 = v_{q+1}, v_{q+2}, ..., v_m$, representando o espaço nulo de S_w, forma a matriz de transformação Z_1, composta dos auto-vetores de $P_1^T S_b P_1$. O primeiro vetor discriminante k_1 de YLDA, é dado por $P^1_{ylda} = P_1 Z_1$, em que geralmente $k_1 = c - 1$;
4. A segunda matriz de projeção $P_2 = v_1, v_2, ..., v_q$, forma a matriz de transformação Z_2, composta dos auto-vetores correspondentes a k_2 referente ao maiores auto-valores de $(P_2^T S_w P_2)^{-1}(P_2^T S_b P_2)$. O vetor discriminante k_2 de YLDA é dado por $P^2_{ylda} = P_2 Z_2$, em que k_2 é um parâmetro de entrada que pode ser extendido ao número final de atributos de LDA, além de ter (c - 1) auto-valores não zeros;
5. Através disso é formada a matriz de projeção P_{ylda}, que é dada pela concatenação de P^1_{ylda} e P^2_{ylda};

4.5 Soluções Não Lineares

4.5.1 Análise Discriminante Kernel

A idéia básica da técnica de análise discriminante de Fisher baseada na técnica de kernel(KFDA) está em resolver o problema da técnica FLDA tradicional para um espaço de características implícito F que é construído utilizando a regra de Kernel na fórmula 4.5.1.

$$\phi : x \epsilon R^n \rightarrow \phi(x) \epsilon F. \tag{4.5.1}$$

A principal característica da técnica de Kernel está em que o vetor de atributos implícito ϕ, não precisa ser computado explicitamente, isso quando o produto de qualquer dois vetores em F seja computado baseado na função de Kernel [37].

A técnica de Kernel, de forma similar a técnica tradicional LDA, é baseada na busca por projeções que maximizam a matriz de dispersão inter-classe e minimiza a matriz de dispersão intra-classe. Dessa forma as matrizes de dispersão inter-classe S_b e intra-classe S_w, sobre o conjunto de amostra F, são computados da mesma forma descrita nas fórmulas 4.1.2 e 4.1.1 respectivamente. Mas ao mesmo tempo cada amostra x_j é substituída por uma função de $\phi(x_j)$

Algoritmo 5: Algoritmo YLDA

Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.

Saída: Matriz Reduzida $YLDA$

$Sw \longleftarrow calculaCovarianciaIntraClasse(amostras, M)$;
$Ev \longleftarrow Sw.analisaAutoValoresAutoVetores()$;
$autoVetores \longleftarrow Ev.autoVetores()$;
$autoValores \longleftarrow Ev.autoValores()$;
$PCA \longleftarrow buscaAutoVetoresMaiorGrauAutoValores(autoVetores, autoValores, R)$;
$PCAAmostras \longleftarrow PCA.transposta().multiplica(amostras)$;
$Sw \longleftarrow calculaCovarianciaIntraClasse(PCAAmostras, M)$;
$Sb \longleftarrow calculaCovarianciaInterClasse(PCAAmostras, M)$;
$Ev \longleftarrow Sw.analisaAutoValoresAutoVetores()$;
$V \longleftarrow Ev.autoVetores()$;
$A \longleftarrow Ev.autoValores()$;
$q \longleftarrow Ev.autoValoresNaoNulos()$;
$P1 \longleftarrow V(q+1, M)$;
$P2 \longleftarrow V(0, q)$;
$AUXp1 \longleftarrow P1.transposta().multiplica(Sb).multiplica(P1)$;
$Ev \longleftarrow AUXp1.analisaAutoValoresAutoVetores()$;
$Z1 \longleftarrow Ev.autVetores()$;
$A \longleftarrow Ev.autoValores()$;
$P1LDA \longleftarrow P1.multiplica(Z1)$;
$AUXp2 \longleftarrow P2.tranposta().multiplica(Sw.multiplica(P2))$;
$AUXp2 \longleftarrow AUXp2.multiplica(P2.transposta().multiplica(Sb.multiplica(P2)))$;
$Ev \longleftarrow AUXp2.analisaAutoValoresAutoVetores()$;
$Z2 \longleftarrow Ev.autoVetores()$;
$A \longleftarrow Ev.autoValores()$;
$P2LDA \longleftarrow P2.multiplica(Z2)$;
$PYLDA \longleftarrow concatena(2, P1LDA, P2LDA)$;
$YLDA \longleftarrow buscaAutoVetoresMaiorGrauAutoValores(real(PYLDA), real(diag(PYLDA)), R)$;
retorna *matrizReduzida(amostras,YLDA)*

das amostras do conjunto F. Consideramos então que o cálculo LDA está implícito sobre o espaço de atributos F. Isso implica que qualquer solução de w em F deve tentar alcançar todas as amostras em F, para qualquer coeficiente $\alpha_i, i = 1, 2, ..., N$, tal que,

$$w = \sum_{i=1}^{N} \alpha_i \phi_i. \tag{4.5.2}$$

Substituindo w pela critério de maximização de Fisher em 4.5.3, podemos obter uma nova solução para o problema de Fisher na fórmula 4.5.4, em que K_b e K_w são baseados nas novas amostras construídas a partir da projeção sobre a função de Kernel, $\zeta_i = (k(x_1, x_i), k(x_2, x_i), k(x_3, x_i), ..., k(x_N, x_i))^T, 1 \leq i \leq N$.

$$J(W) = arg_w max \frac{W^T S_b W}{W^T S_w W}. \tag{4.5.3}$$

$$J(\alpha) = arg_\alpha max \frac{\alpha^T K_b \alpha}{\alpha^T K_w \alpha}, \tag{4.5.4}$$

Uma função de Kernel bem conhecida é a função ilustrada no trabalho [18], conhecida também como $Cosine - Kernel$. Essa função é definida na fórmula 4.5.5.

$$k(x, y) = (\phi(x).\phi(y)) = (a(x.y) + b)^d, \tag{4.5.5}$$

Como valores para os parâmetro da função de $CosineKernel$, muitos experimentos, adotam ($a = 10^{-3}/tamanho_{imagem}$), b=0 e d=2. Esses valores mostraram um bom desempenho na performance para aplicação de reconhecimento de faces. O algoritmo 6 ilustra a técnica KLDA.

4.6 Exemplo Ilustrativo

Para ilustrarmos melhor a funcionalidade de cada técnica aqui apresentada para o problema de singularidade na redução de atributos, iremos simular alguns problemas, que irão nos auxiliar à encontrar as melhores projeções que consigam manter a discriminalidade entre as classes. Para isso separamos 3 exemplos de bases de aprendizagem, que são muito utilizados para demonstrar a eficiência de técnicas de redução de atributos.

Cada exemplo apresentado irá conter casos específicos de dispersão de dados entre as classes e entre suas amostras, dificultando assim a redução de atributos. Para cada problema utilizaremos dois atributos, sendo eles nomeados de x e y, sendo representados por 3 classes, chamadas de $classe1$, $classe2$ e $classe3$.

O primeiro exemplo, ilustrado na imagem 4.4(a), é baseado em um conjunto de amostras mais simples, onde as classes para classificação estão consideravelmente dispersas e que visivelmente é possível identificar uma projeção que consiga manter a discriminação entre as classes.

O segundo exemplo, ilustrado na imagem 4.4(c), já se trata de uma base um pouco mais complexa. Nesse exemplo se quisermos utilizar o eixo x como nosso atributo reduzido teremos uma leve sobreposição das amostras contidas na $classe2$ e $classe3$, ou se projetarmos as amostras para o eixo y, iremos sobrescrever quase que por completamente a $classe1$ com a $classe3$.

Algoritmo 6: Algoritmo KLDA

Entrada: $(amostras)$ = Conjuntos de Amostras, (M) = Dimensão Original, (R) = Dimensão a ser reduzida.

Saída: Matriz Reduzida $KLDA$

```
1  para i ← 1 até tamanho(listaClassesAmostras) faça
2      vetores ⟵ listaClassesAmostras(i);
3      posicaoColunaKernel ⟵ 0;
4      para j ← 1 até tamanhoLinhas(vetores) faça
5          vetor ⟵ vetoresLinha(j);
6          para c1 ← 1 até tamanho(listaClassesAmostras) faça
7              vetorZeta ⟵ listaClassesAmostras(c1);
8              para c2 ← 1 até tamanhoLinhas(vetorZeta) faça
9                  vetorInterno ⟵ vetorLinhaZeta(c2);
10                 valor ⟵ ((a ∗ (vetor ∗ vetorInterno′) + b)^d);
11                 kernel.setarValor(posicaoLinhaKernel, posicaoColunaKernel, valor);
12                 posicaoColunaKernel + +;
13         posicaoLinhaKernel + +;
14         posicaoColunaKernel ⟵ 0;
15     posicaoLinhaKernel ⟵ 0;
16     amostras(i) ⟵ kernel;
17 retorna FLDA(amostras,M,R)
```

No terceiro exemplo, Figura 4.4(e), é ilustrado um caso problemático para a redução de atributos, em que temos as três classes de amostras muito próximas uma da outra, e com uma variância baixa das amostras. Um caso como esse, poderia influenciar diretamente na presença da singularidade quando fosse seguido o critério de Fisher. Nas Figuras 4.4(b), 4.4(d) e 4.4(f), podemos visualizar os resultados de projeções para cada exemplo.

A partir dos resultados de cada projeção conseguimos verificar que para os três problemas de amostras, conseguimos encontrar projeções que conseguem manter a discriminalidade entre as classe de classificação. Para o primeiro conjunto de amostras ilustrado na imagem 4.4(b), podemos verificar que as projeções das técnica de DLDA, YLDA, FisherFace e Kernel, se aproximam muito uma da outra, diferente da técnica CLDA que possui uma projeção mais próxima do eixo y.

No segundo conjunto de amostras na imagem 4.4(d), temos uma caso bem interessante, em que as técnicas CLDA e DLDA, possuem resultados próximos um do outro, semelhante também com os resultados de projeções das técnicas YLDA, FisherFace e Kernel.

No terceiro conjunto na imagem 4.4(f), mesmo tendo um caso mais complexo de avaliação e para encontrar uma melhor projeção, podemos considerar que as técnicas tiveram um bom resultado, em que todas tenderam para uma mesma direção, fazendo com que a partir das projeção das amostras sobre o eixos de classificação, teremos uma pequena sobreposição de amostras da $classe1$ sobre as amostras da $classe2$ e $classe3$.

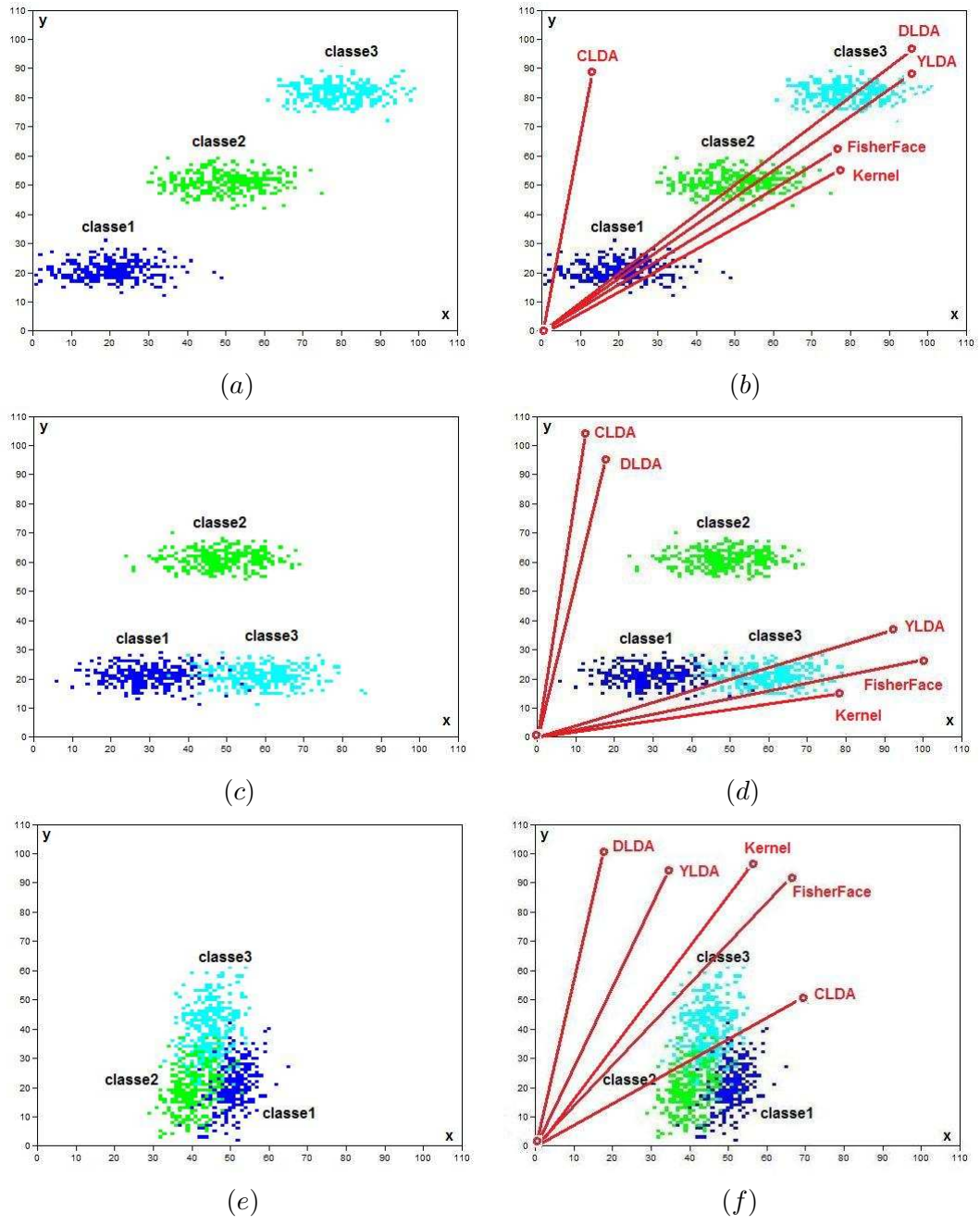


Figura 4.4: Amostras para a redução de atributos. (a), (c) e (e) Exemplo de amostras. (b), (d) e (f) Redução de seu respectivo conjunto de amostras.

Capítulo 5

Desenvolvimento

Nesse trabalho foram realizados diversos experimentos com objetivo de identificar e solucionar os problemas de singularidade utilizando Análise Discriminante de Fisher. Além das soluções que foram implementadas criamos também um ambiente para a geração de experimentos na aplicação da detecção automática de defeitos em couro bovino.

Para a criação desse ambiente foram desenvolvidos quatro módulos. O primeiro módulo é responsável em criar um ambiente para armazenar imagens e realizar a marcação e rotulação de regiões de interesse, independente de qual aplicação. Com essas imagens armazenadas e rotuladas o módulo é capaz de realizar a geração de amostras referente às marcações realizadas. Esse módulo poderá assim resolver futuros problemas, como a baixa quantidade de imagens para experimentos.

O segundo módulo implementado tem como objetivo criar um ambiente para a geração de experimentos, dando opções de extração de atributos e a criação de arquivos de base de aprendizagem no formato de interpretação para experimentos no *WEKA*, para que possamos assim realizar testes com diferentes classificadores. Esse módulo é responsável em criar a base de treinamento (datasets) para a aprendizagem, passando essas informações para o próximo módulo.

As funcionalidades do terceiro módulo estão embutidas sobre as do segundo módulo. Esse módulo utiliza as implementações como a técnica de Fisher e suas variações, para a redução de atributos e classificação automática. Com isso, esse módulo captura a base de aprendizagem gerada pelo segundo módulo, para reduzir o conjunto de atributos, gerando uma nova base reduzida.

O quarto módulo é uma implementação mais específica para o objetivo geral do projeto DTCOURO, que é o resultado da classificação do couro. Com esse módulo é possível a criação de regras de classificação para determinar a faixa de valores que serão utilizadas para a classificação geral do couro, como: Classe A, Classe B, Classe C, ou refugo.

5.1 Ferramentas de Apoio

Para a implementação desses quatro módulos utilizamos três ferramentas livres, que são: *IMAGEJ*, *WEKA* e *SIGUS*. A seguir serão apresentadas mais informações sobre os módulos desenvolvidos e as ferramentas utilizadas.

5.1.1 IMAGEJ

Para a utilização de recursos de processamento digital de imagens e visão computacional, foi utilizado do pacote *ImageJ*, que é uma versão multiplataforma do software *NIH* Image, para *Macintosh*. Entre os recursos oferecidos pelo pacote, destacamos a disponibilidade de programas-fonte abertos de diversos algoritmos como: manipulação dos mais variados formatos de arquivo de imagens, detecção de bordas, melhoria de imagens, cálculos diversos (áreas, médias, centróides) e operações morfológicas. Esse software disponibiliza também um ambiente gráfico que simplifica a utilização de tais recursos, além de permitir a extensão através de plugins escritos em Java. Outro fator importante para a sua utilização é a existência de uma grande comunidade de programadores trabalhando em seu desenvolvimento com novos plugins sendo disponibilizados freqüentemente [24].

5.1.2 WEKA

O segundo pacote proposto para a utilização no desenvolvimento dos módulos e realização dos experimentos de classificação foi o *WEKA* (Waikato Environment for Knowledge Analysis), também escrito em Java e com programas-fonte abertos. O *WEKA* é um ambiente bastante utilizado em pesquisas na área de aprendizagem de máquina, pois oferece diversos componentes que facilita a implementação de classificadores. Além disto, esse ambiente permite que novos algoritmos sejam comparados a outros já consolidados na área de aprendizagem, como o C4.5, o Backpropagation, o KNN e o naiveBayes, entre outros [24]. Esse pacote foi utilizado para auxiliar nos testes de classificação que serão realizados sobre a base de aprendizagem gerada pelos módulos. Com essa ferramenta foi possível também obter facilmente dados como: tempo de aprendizagem, tempo de classificação, porcentagem de acerto, porcentagem de erro, falso positivo, falso negativo, matriz de confusão, entre outros.

5.1.3 SIGUS

O terceiro pacote utilizado para auxiliar as implementações desse trabalho foi o *SIGUS*. A plataforma *SIGUS* é um ambiente computacional de apoio ao desenvolvimento de sistema para inclusão digital de pessoas com necessidades especiais, com objetivo de aumentar a quantidade de programas destinados a essas pessoas [22]. Através deste pacote, foi possível o uso de diversas técnicas de extração de atributos, as quais utilizamos: matriz de co-ocorrência, mapas de interação, atributos de cores (HSB) e (RGB) e Filtros de Gabor.

5.2 Ambiente de Aquisição e Marcação

O primeiro módulo implementado auxilia nas seguintes fases de processamento: (1) fase de aquisição de imagens, (2) fase de pré-processamento, (3) fase de marcação e rotulação das principais áreas que concentram os defeitos e a (4) fase de captura de amostras para testes. Na fase de aquisição de imagens, foi criada uma janela de seleção e inserção, para que possam ser cadastradas e pré-visualizadas imagens que servirão como base de trabalho. A janela também dá opções de visualização das amostras geradas e as regiões de marcação. Na fase de pré-processamento o módulo tem a opção de realizar processamentos nas imagens como: filtros,

limiarização, detector de bordas, escolha de níveis de cinza como 8,16,32 bit, conversão de formato de cores como RGB e HSB, entre outras.

Na fase de marcação e rotulação é possível realizar marcações sobre regiões de interesse. A cada marcação, o módulo cria um registro no banco de imagens, contendo informações capturadas, como: nome do defeito marcado, tipo de defeito selecionado e região da marcação. Em qualquer marcação realizada sobre a imagem, o módulo captura todas as informações referente a marcação, como: tipo de marcação e a região em pixels da região marcada. Todas essas informações são armazenadas automaticamente a cada marcação, criando assim uma máscara ou camada sem modificação na imagem original. Dessa forma o módulo irá realizar a marcação visual a medida que a imagem é solicitada, caso isso ocorra, o módulo buscará no banco de dados todas as informações referentes as marcações.

Na fase de captura de imagens de testes, o módulo utiliza a base de imagens marcadas e armazenadas no banco de dados. Para a geração dessas amostras foi criada uma técnica baseada em uma varredura com espaçamento uniforme. Esse modo de geração tem como objetivo principal varrer a imagem ativa de modo uniforme, através das informações capturadas do usuário como: marcação selecionada, quantidade de amostras, largura e comprimento em pixel de cada amostra, percorrer a imagem e verificando os quatro pontos de cada amostra se estão localizados dentro da marcação selecionada. Caso esteja localizado dentro da região, o modo varredura cria uma cópia dessa amostra e armazena sua localização em uma lista de possibilidades e caso não esteja localizado dentro da região, o sistema irá ignorar, e continuará sua varredura até o fim da imagem. O algoritmo 7 realiza a geração de amostras.

Algoritmo 7: Algoritmo do gerador de amostras com espaçamento uniforme.

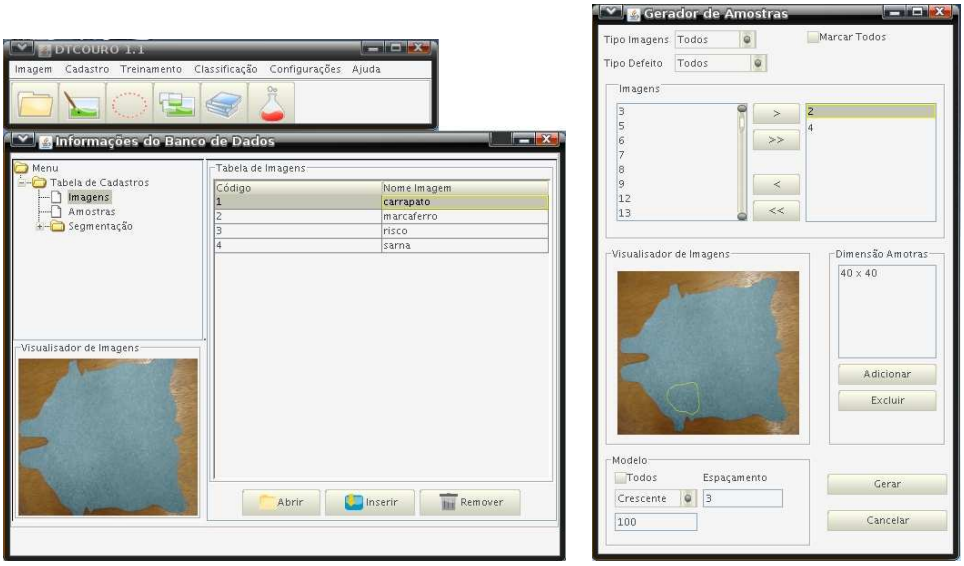
Entrada: ($listaDefeitos$) = Lista de tipo de defeitos, ($quantidadeAmostras$) = Quantidade de amostras por defeito.
Saída: Amostras geradas.

```
1 para i ← 1 até tamanho(listaDefeitos) faça
2     listaMarcacao ⟵ buscarMarcacoes(listaDefeitos(i));
3     para j ← 1 até tamanho(listaMarcacao) faça
4         listaPossibilidades ⟵ buscaePossibilidades(listaMarcacao(j));
5         amostrasGeradas ⟵
          gravarMarcacao(listaPossibilidades, quantidadeAmostras);
6 retorna amostrasGeradas
```

A quantidade total de possibilidades para captura das amostras é salva e depois é realizada uma divisão sobre o número total de amostras que se deseja gerar. Esse valor é o incremento utilizado para listar as possibilidades geradas para assim gravar suas informações no banco de dados. Com isso, o módulo cria um banco de amostras específicas para cada tipo de defeito selecionado e com um modelo uniforme garante uma base de aprendizagem mais complexa e real para a classificação automática. A Figura 5.1 ilustra a janela do primeiro módulo implementado.

5.3 Módulo de Geração de Experimentos

O módulo de geração de experimentos foi criado com o objetivo de facilitar o uso desde a manipulação de amostras, extração de atributos e a geração de bases de aprendizagem que serão



(*a*) (*b*)

Figura 5.1: (a) Tela de Aquisição de Imagens para marcação e rotulação e (b) Tela do gerador de amostras.

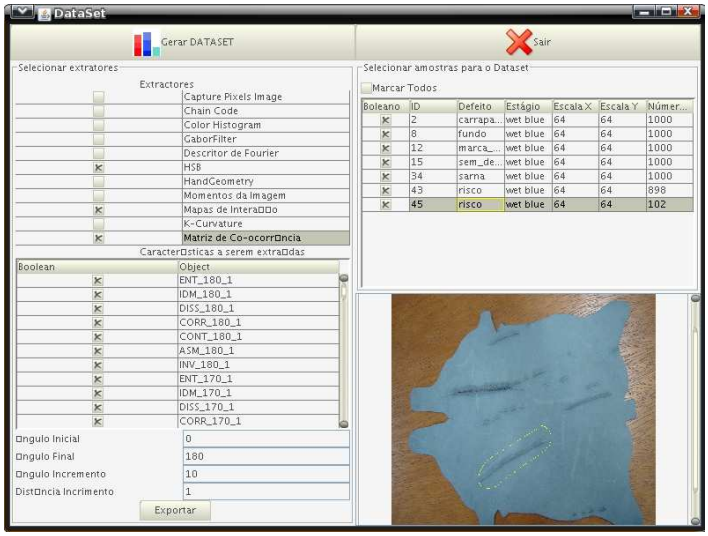
utilizadas no módulo de redução de atributos e classificação automática.

Esse ambiente pode ser dividido em duas seções: a seção de seleção de amostras e a seção de extração de atributos e geração da base de aprendizagem. A seção de seleção de amostras tem somente a opção de escolha dos tipos de amostras que representarão as classes de defeitos para classificação. A partir das amostras selecionadas, é possível então utilizar a seção de extração de atributos, que disponibiliza uma lista com todos os tipos de métodos de extração, localizados dentro da biblioteca *SIGUS*. Dessa forma, a partir das amostras selecionadas e dos métodos escolhidos de extração de atributos, já é possível a geração da base de aprendizagem no formato do arquivo de interpretação do *WEKA*, extensão ".arff". A Figura 5.2 ilustra a janela do segundo módulo implementado.

5.4 Módulo de Redução de Atributos e Classificação Automática

A criação do módulo de redução de atributos e classificação automática teve como finalidade a realização da classificação automática dos defeitos encontrados com a nova base de aprendizagem criada, a partir da redução de atributos utilizando aqui as técnicas implementadas, baseadas em LDA. A partir disso, o módulo é capaz de varrer a imagem ativa, extraindo os atributos selecionados, reduzindo e utilizando a nova base de aprendizagem gerada para a classificação automática. Foi desenvolvido também utilizando a biblioteca *WEKA*, a opção de escolha do algoritmo de aprendizagem que será utilizado na classificação, como: árvore de decisão, Naive Bayes, SVM, redes neurais, entre outros.

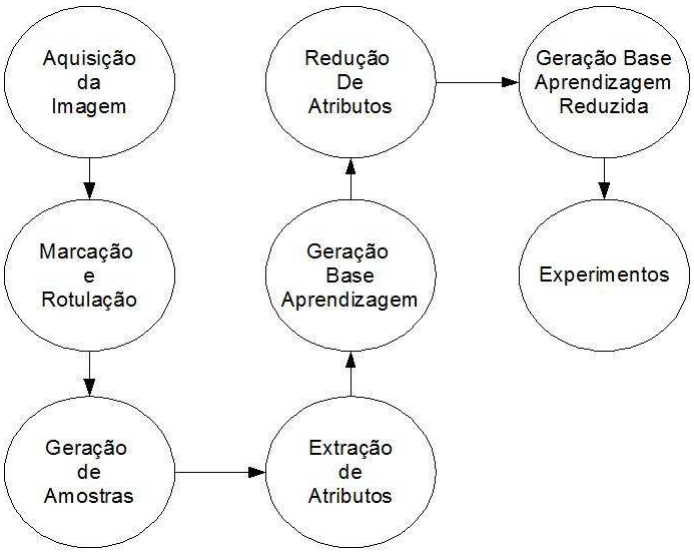
A implementação das técnicas de redução de atributos baseadas em LDA foi realizada em duas fases. A primeira fase foi a implementação da técnica utilizando uma ferramenta chamada



(*a*)

Figura 5.2: Tela do segundo módulo para a criação de bases de aprendizagem.

Scilab. Em seguida, realizamos a execução da segunda fase, que foi a implementação em *Java* em forma de plugin para o projeto Sigus. O objetivo da implementação da primeira fase, foi criar inicialmente um ambiente de testes matemáticos, utilizando uma linguagem de alto nível de programação. A partir da primeira fase então, pôde ser possível implementar facilmente a segundo fase, que foi utilizada para a realização dos experimentos e testes, para encontrarmos as melhores soluções de redução de atributos, para casos que ocorrem ou não problemas de singularidade. O Diagrama 5.3 ilustra como foi realizado esse processo.



(*a*)

Figura 5.3: Modelo de interação dos quatro módulos implementados.

5.5 Módulo de Regras de Classificação

A implementação do módulo de criação de regras de classificação teve como objetivo facilitar experimentos, em que era necessário termos uma classificação geral do couro bovino, utilizando o classificador automático desenvolvido.

Como esse módulo podemos setar valores tolerados ou não, para cada tipo de defeito e para cada classe geral de classificação, como, Classe A, Classe B e Classe D, ou refugo. Dentro de cada classificação, é possível fixar valores de porcentagem representando assim o valor máximo do defeito específico tolerado que será utilizado para representar o couro após a classificação automática pelo sistema.

Esse módulo irá ajudar a diversos tipos de usuários do sistema DTCOURO, principalmente da Embrapa, a realizar classificações no couro bovino, comparando resultados de classificações para diferentes áreas de uso. Uso exemplo disso é a classificação do couro bovino para fabricantes de estofados, que é diferente da classificação para fabricante de sapatos. A Figura 5.4, ilustra a tela de criação de regras para a classificação.

Figura 5.4: Tela para criação de regras para classificação.

Capítulo 6

Experimentos, Resultados e Análise

Nesta seção serão descritos todos os detalhes sobre os experimentos realizados utilizando a técnica tradicional de Análise Discriminante de Fisher e as técnicas que apresentamos como soluções para o problema de singularidade, a fim de realizarmos análises de seus resultados de redução de atributos e classificação automática. Para a realização dos experimentos dos módulos implementadas foi utilizado um computador com processador Pentium Core 2 Duo - 2.3 GHz com 2 GB de memória RAM e sistema operacional Windows Vista.

6.1 Equipamento

Durante as viagens para a captura das imagens nos frigoríficos e curtumes que foram utilizadas em nossos experimentos, foi utilizado um equipamento portátil de nível de filmagens. Esse equipamento é composto por uma câmera digital de 14 mega pixels, monitor Lcd 4 polegadas e uma mini-grua modelo $SMC-4$. A mini-grua é composta por uma lança de 1600 mm e tripé, fazendo com que haja um mecanismo que mantém constante a direção da câmera acoplada em sua cabeça durante movimentos de captura de filmagens. As Figuras 6.1(a) e 6.1(b) ilustram o equipamento montado e modo da captura das imagens.

Acoplamos também o monitor ao tripé e à câmera digital, utilizando assim como interface para captura de imagens mais complexas, como por exemplo, defeitos que se encontram no meio do couro bovino, o que sem isso seria impossível um padrão de captura dessas imagens. Com isso conseguimos não só maior qualidade nas imagens, mas também diminuir influências de características como iluminação e variação da distância na captura de imagens sobre couros bovinos. As Figuras 6.2(a) e 6.2(b) ilustram dois tipos de áreas para captura das imagens. Para esse trabalho utilizamos área com iluminação natural.

6.2 Procedimentos

Junto com as pesquisadoras Mariana de Aragão Pereira, Alexandra Oliveira e do pesquisador Manoel Jacinto da EMBRAPA e acadêmicos do grupo de pesquisa em Engenharia e Computação (GPEC) foi criado um banco de imagens, com diferentes tipos de defeitos em couro cru e no estágio de curtimento Wet-Blue.

(*a*) (*b*)

Figura 6.1: Modelo projetado para captura das imagens. (a) tripé montato, (b) Posicionamento para captura das imagens.

(*a*) (*b*)

Figura 6.2: Àreas para capturas de imagens. (a) iluminação artificial, (b) iluminação natual.

Todas as imagens aqui utilizadas foram pré-cadastradas e em seguida, para cada uma foram realizadas as marcações, com ajuda de especialistas, referentes aos seus específicos defeitos. Com essas informações armazenadas no banco de dados foi possível então realizar a fase de geração de amostras. Nessa fase o módulo utilizou todos os dados das marcações para realizar uma varredura sobre os tipos de defeitos cadastrados, criando assim um conjunto de amostras específicas.

Como esse trabalho tem o objetivo de avaliar o desempenho da redução de atributos e das técnicas utilizando análise discriminante, utilizamos todas as imagens coletadas nos estágios de couro cru e Wet-Blue. Para a realização de todos os experimentos aqui presentes foram utilizadas 20 imagens do couro cru e 50 imagens do couro bovino no estágio Wet-Blue, sendo elas portadoras de defeitos: marca de ferro, risco, sarna, carrapato, esfola, berne, furo e também imagens sem nenhum defeito. Todas as imagens aqui utilizadas são de animais pertencentes as raças Nelore e Hereford.

Nesse trabalho foram realizados diversos experimentos utilizando vários tipos de defeitos de estágios do couro bovino os quais são mais conhecidos e utilizados na classificação por um especialista, com base nos curtumes e frigoríficos visitados. Dessa forma organizamos os experimentos da seguinte forma.

1. Classificação de defeitos do couro bovino no estágio Cru.
2. Classificação de defeitos do couro bovino no estágio Wet-Blue.
3. Redução de atributos sobre imagens com defeitos no Couro-Cru utilizando a técnica tradicional de Análise Discriminante de Fisher;
4. Redução de atributos sobre imagens com defeitos no Wet-Blue utilizando a técnica tradicional de Análise Discriminante de Fisher;
5. Experimento de simulação para o Problema de Singularidade;
6. Redução atributos baseadas em LDA, utilizando imagens com defeitos sobre o Couro-Cru e Wet-Blue;
7. Classificação automática de defeitos em comparação à classificação por especialistas na área.

6.3 Imagens utilizadas para os Experimentos

As imagens utilizadas para os experimentos foram coletadas durantes viagens realizadas junto com a equipe da Embrapa. Essas viagens foram realizadas e divididas por estado, curtume e frigorífico, com o objetivo de realizar a classificação desde o abate do boi até após o seu curtimento Wet-Blue.

Para a coleta dessas imagens foi visitado para os nossos experimentos, o curtume da Bertin, localizado na cidade de Lins-SP, sendo conhecido como um dos maiores curtumes brasileiros. A escolha desse curtume foi baseada em informações como localização do destino a qual o couro classificado na fase de abate seria transportado.

Durante essa viagem foram obtidas ao todo 150 imagens do couro Wet-Blue sendo elas pertencentes a 31 bovinos. Essas imagens foram tiradas com uma altura fixa padronizada pelo

equipamento de 1metro e 10cm. Essa altura foi selecionada com base em testes em campo, pois para uma altura menor, não teríamos informações de vários defeitos em conjunto e uma altura maior seria muito difícil a visualização pelo sistema na fase de marcação para a geração de amostras.

Para a captura de imagens do couro cru foram encontrados vários problemas durante sua execução, tais como: dificuldade de acesso, dificuldade de filmagens do local, ambiente controlado, entre outros. Ao contrário dos curtumes, onde o ambiente é natural, fácil acesso a regiões que se encontra o couro e um grande apoio em sua classificação. Com isso não conseguimos muitas imagens para a realização dos experimentos na fase do couro cru, conseguindo somente 20 imagens, contendo os mesmos defeitos que o couro Wet-Blue.

6.4 Marcação e Geração de Amostras

A partir do módulo de marcação foram realizadas para cada imagem, três formas de marcação, que foram: defeito específico da imagem, região sem defeito, e região ou objeto que se encontra no couro bovino. O principal objetivo para esse tipo de regra de marcação foi a necessidade, de se obter uma maior variação de regiões sem defeito, fundo e regiões com defeitos, próximas uma da outra.

Durante a fase de marcação houve uma grande preocupação com os defeitos que são visualmente similares, isto é, defeitos que possuem características visuais semelhantes, como risco e esfola, para que quando fossem realizados os testes, não tivessem tantos efeitos em sua classificação. A Figura 6.3 ilustra duas amostras com exemplo de semelhanças para o couro cru e couro wet-blue.

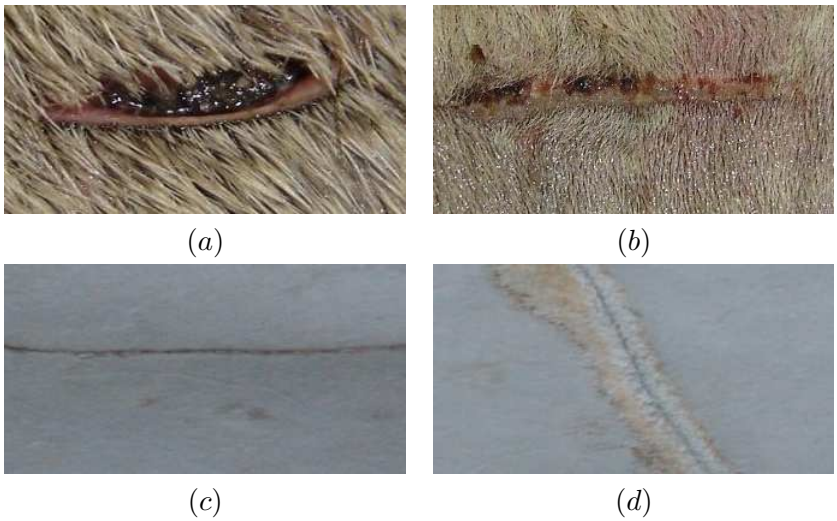
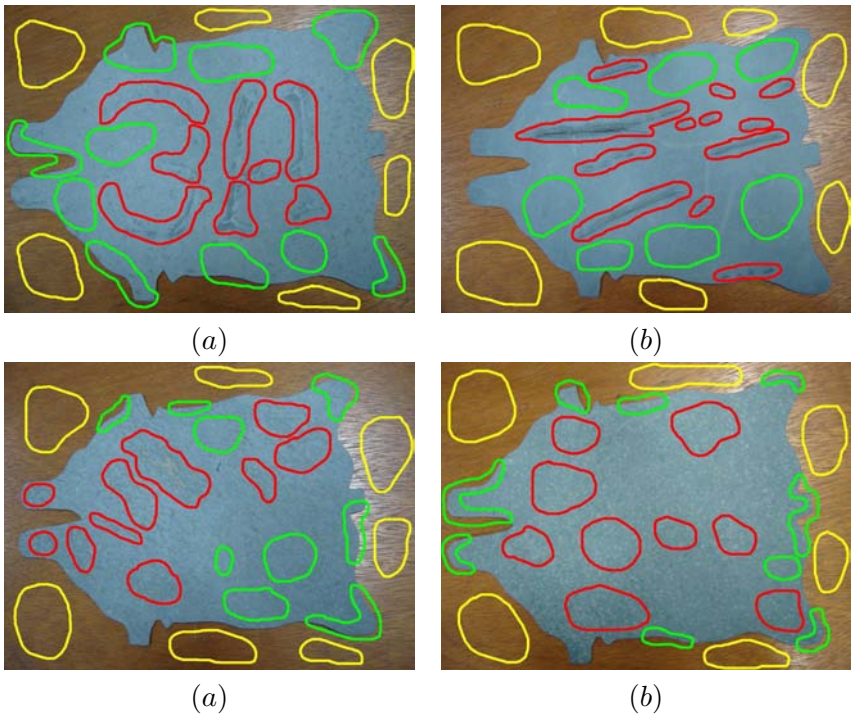


Figura 6.3: Exemplo de amostras semelhantes. (a) e (c) Amostras com defeito de Risco, (b) e (d) Amostras com defeito de esfola.

Dessa forma, a marcação foi realizada especificamente na parte do defeito que mais se destacava, possibilitando assim uma melhora na base de dados do classificador. Em seguida, para cada marcação cadastrada no banco de dados foram geradas na escala de 40x40 pixel, 2.000 amostras para cada tipo de defeito, totalizando 16.000 amostras para Wet-Blue e 16.000

amostras para o couro-cru. A diferença das amostras geradas pelo couro cru e Wet-Blue, esta na quantidade de marcações utilizadas para captura dessas amostras, em que para o Wet-Blue foram realizadas 60 marcações e o couro cru apenas 30 marcações, isso devido à baixa quantidade de imagens registradas.

Escolhemos a quantidade de 2.000 amostras por que em experimentos preliminares verificamos que essa quantidade apresentou bom resultado de classificação e também um tempo considerável na redução de atributos. Para bases maiores tivemos problemas com o tempo de processamento para redução dos atributos. Como o objetivo do nosso trabalho é avaliar o desempenho quanto a redução acreditamos que mantendo uma quantidade fixa de amostras para cada defeito, mas também garantirmos que essas amostras fossem capturadas de todas as marcações realizadas e de modo aleatório e uniforme poderíamos ter uma base de aprendizagem mais similar com a realidade do problema. A Figura 6.4 ilustra exemplos de como foi realizada as marcações manualmente.



(a) (b)

(a) (b)

Figura 6.4: Exemplo de imagens marcadas manualmente pelo módulo de aquisição e marcação de imagens, com as seguintes cores, (vermelho = defeito, amarelo = fundo e verde = sem defeito). (a) defeito marca ferro, (b) defeito risco, (c) defeito sarna e (d) defeito carrapato.

6.5 Classificação de defeitos do couro bovino no estágio Cru

Esse primeiro experimento foi realizado com o objetivo de analisar inicialmente o desempenho dos métodos de extração de atributos quanto à classificação dos tipos de defeitos aqui pré-selecionados do couro bovino no estágio couro-cru, sem a redução de atributos. Os métodos foram selecionados com base em experimentos como [25], que mostraram que técnicas com base em análise de níveis de cinza e atributos relacionados a modelos de cores, podem apresentar

ganhos satisfatórios no desempenho quanto a classificação de imagens de defeitos em couro.

Esse tipo de experimento sobre o couro-cru é muito importante, pelo fato de ser a fase de análise do couro onde temos mais informações visíveis sobre os vários tipos de defeitos. Com essa primeira parte do experimento poderemos iniciar nossa análise verificando se a taxa de classificação correta será satisfatória.

A partir de cada amostra gerada dos defeitos marcados, o módulo de geração de experimentos realizou a extração de atributos. Os atributos capturados pelo módulo foram: matriz de co-ocorrência com variações de 0º a 180º com intervalo de 10º e distância de 1 pixel extraindo 126 atributos, mapas de interação com variações de 0º a 180º com intervalo de 10º e distância inicial de 0 a 2 com intervalo de 1 pixel extraindo 7 atributos, filtros de gabor com parâmetros de tamanho da onda (100:256), orientação (45:135), simetria (90:90), tamanho núcleo de 1.0 e excentricidade 0.5 extraindo 15 atributos, valores médio do histograma de cada componente de cor de HSB e RGB com 6 atributos e mais 6 atributos dos valores discretizados em 3 intervalos de HSB e RGB. Para cada matriz gerada da técnica de matriz de co-ocorrência e mapas de interação, foram capturados os atributos: entropia (ENT), momento da diferença inversa (IDM), dissimilaridade ($DISS$), correlação ($CORR$), contraste ($CONT$), segundo momento angular (ASM) e a diferença inversa (INV), totalizando 148 atributos de textura e 12 atributos de cor. A Tabela 6.1 resume os atributos extraídos.

Tabela 6.1: Atributos extraídos para o experimento de classificação de defeitos do couro-cru

Método de Extração	**Quantidade de Atributos**
Média (H), (S), e (B)	3
Média (R), (G) e (B)	3
HSB (Discretizado em 3 intervalos)	3
RGB (Discretizado em 3 intervalos)	3
Mapas de Interação	7
Matriz de Co-ocorrência	126
Filtros de Gabor	15
Total	**160**

Com base nos atributos extraídos utilizamos para realizar os experimentos de classificação, os algoritmos: C4.5, SMO, IBk e NaiveBayes. O motivo que nos levou a escolher esses classificadores foi que em experimentos correlatos [23] e [25], essas técnicas apresentaram os melhores resultados de classificação. Como configuração dos parâmetros de cada classificador utilizamos os valores padrões setados inicialmente pelo Weka. Os valores dos parâmetros de cada técnica aqui utilizada podem ser visualizados nas tabelas 6.2, 6.3, 6.4 e 6.5. Escolhemos como modo de testes dos resultados, a técnica estatística de validação cruzada com configuração de 3 dobras e 2 validações. Para a visualização dos resultados utilizamos a taxa(%) de classificação correta.

6.5.1 Resultados e Análise

Os resultados de classificação das imagens do couro no estágio Couro Cru, referente a cada técnica de aprendizagem são apresentados na Tabela 6.6. Podemos concluir pelo resultado de classificação que todos os classificadores se comportaram de maneira eficiente, menos a técnica de $NaiveBayes$. A técnica de classificação que mais se destacou foi a IBk chegando no melhor

Tabela 6.2: Parâmetros setados para o classificador C4.5

Parâmetro	**Valor**
Separações Binárias	Falso
Fator de Confiança	0.25
Número mínimo de objetos	2
Número de dobras	3
Poda de Redução de Erro	Falso
Semente	1
Crescimento de Sub-Árvores	Verdadeiro
Sem Poda	Falso
Usar Laplace	Falso

Tabela 6.3: Parâmetros setados para o classificador IBk

Parâmetro	**Valor**
KNN	1
Validação cruzada	Falso
Distância de ponderação	Nenhuma
Quadrado Médio	Falso
Sem normalização	Falso
Tamanho janela	0

Tabela 6.4: Parâmetros setados para o classificador SMO

Parâmetro	**Valor**
Contruír logística	Falso
c	1.0
Tamanho cache	250007
Epsilon	1.0E-12
Expoente	1.0
Espaço de atributos	Falso
Tipo de filtro	Normalizar
Gamma	0.01
Menor ordem	Falso
Número de dobras	-1
Semente Aleatória	1
Parâmetro de tolerância	0.0010
Usar RBF	Falso

Tabela 6.5: Parâmetros setados para o classificador NaiveBayes

Parâmetro	**Valor**
Usar estimador Kernel	Falso
Usar discretização supervisionada	Falso

caso com uma taxa de acerto de 95.90% de classificação correta.

Tabela 6.6: Resultado classificação imagens Couro-Cru

Classificador	**Taxa(%) Acerto**	**Desvio Padrão**
C4.5	94.73	0.13
IBk	95.90	0.16
Naives Bayes	47.37	0.48
SMO	90.03	0.06

A partir desse resultado podemos perceber que dependendo da técnica de classificação, para análise do couro cru poderemos influenciar significativamente no resultado final da classificação automática. Com esses resultados e o resultado a seguir para experimentos com imagens do couro em sua fase $Wet - Blue$ poderemos identificar melhor a influência na escolha de um classificador em um sistema para detecção e reconhecimento automático de padrões no couro bovino.

6.6 Classificação de defeitos do couro bovino no estágio Wet-Blue

O segundo experimento foi realizado com o mesmo objetivo do primeiro sobre imagens do couro cru, que está em analisar o desempenho das técnicas de extração de atributos e técnicas de aprendizagem para a classificação de imagens do couro bovino no estágio Wet-Blue.

Os experimentos de classificação sobre as imagens do couro no processo Wet-Blue são importantes, por ser o principal estágio de processamento do couro bovino após seu abate e antes de sua revenda. Esse tipo de experimento identifica se é possível a discriminação de diferentes tipos de defeitos com essa textura, em que visualmente sua análise é mais complexa na classificação do que o couro cru.

Esse segundo experimento foi realizado da mesma forma que o primeiro, em que cada amostra gerada dos defeitos marcados foi utilizada o módulo de geração de experimentos para a extração de atributos. Os atributos capturados pelo módulo também foram os mesmos, como: matriz de co-ocorrência, mapas de interação, filtros de Gabor, valores médio do histograma de cada componente de cor de HSB e RGB e os valores discretizados em 3 intervalos de HSB e RGB. Para os testes de classificação utilizamos os classificadores: C4.5, SMO, IBk e NaiveBayes e capturamos a taxa(%) de classificação correta, utilizando validação cruzada com configuração de 3 dobras e 2 validações.

6.6.1 Resultados e Análise

Os resultados de classificação para imagens do couro no estágio Wet-Blue pode ser visualizado na tabela 6.7. Da mesma forma que no primeiro experimento conseguimos verificar que as técnicas de extração de atributos e de classificação se comportaram de maneira eficiente, de acordo com seu específico nível de complexidade como quantidade de marcações e imagens utilizadas. Mas com algumas observações, como a técnica $NaiveBayes$ que novamente teve a menor taxa de classificação. O mesmo ocorreu para a técnica IBk, que teve novamente a melhor taxa de classificação.

Um ponto importante também para análise é o desempenho das técnicas de extração de atributos que mesmo com uma quantidade maior de marcações para análise do couro Wet-Blue possibilitou a criação de um padrão de discriminalidade entre as classes, facilitando assim as técnicas de aprendizagem e classificação podendo conseqüentemente influenciar nos resultados de redução de atributos.

Tabela 6.7: Resultado classificação imagens Wet-Blue

Classificador	**Taxa(%) Acerto**	**Desvio Padrão**
C4.5	91.72	0.17
IBk	93.76	0.11
Naives Bayes	54.51	0.53
SMO	84.53	0.18

6.7 Redução de atributos sobre imagens com defeitos no Couro-Cru utilizando a técnica tradicional de Análise Discriminante de Fisher

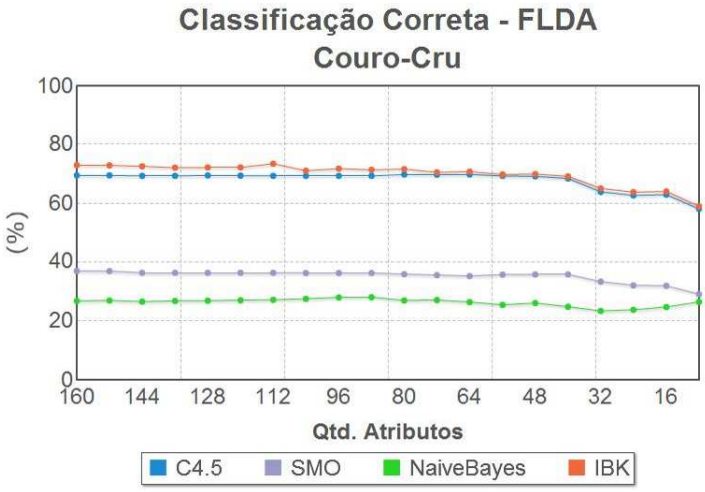
Com base nos resultados de classificação dos experimentos realizados nas seções 6.5 e 6.6, verificamos que existe uma complexidade para a discriminação entre os tipos de defeitos presentes no couro bovino, tanto no couro-cru, quanto no estágio de curtimento Wet-Blue. Mas para alguns classificadores como o IBK, $C4.5$ e SMO, o desempenho na classificação foi satisfatório para um reconhecimento automático de padrões.

Nesse experimento nosso objetivo foi verificar qual o desempenho da classificação de defeitos, a partir da redução de atributos de uma base de aprendizagem. Isso nos ajudou a termos uma breve noção do poder da redução de informações para problemas com uma complexidade maior, em que a quantidade e a qualidade das informações para a discriminação é de suma importância para uma taxa alta de classificação correta.

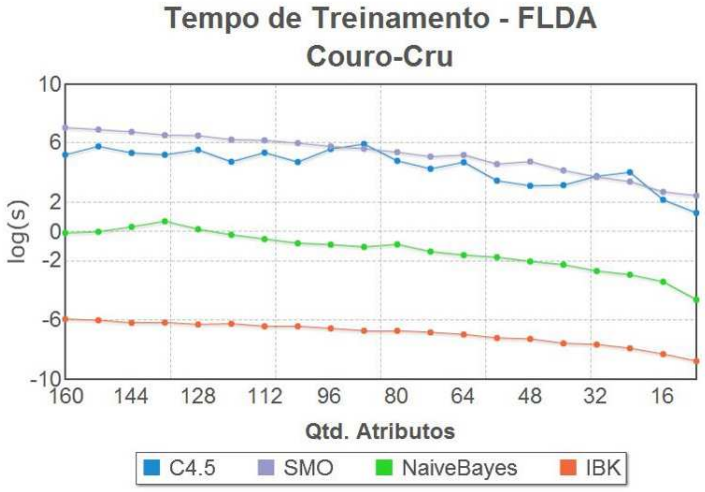
Para esse experimento utilizamos imagens do couro em sua fase crua, sem nenhum pré-processamento. As imagens, os atributos extraídos e técnicas de aprendizagem para a classificação aqui utilizada foram as mesmas que no primeiro experimento.

Com base nos 160 atributos extraídos das amostras e junto com a técnica implementada de Fisher, para redução de atributos foram geradas 20 bases de aprendizagem, cada uma contendo a_i atributos, com i variando de uma porcentagem de 5% decrementada sobre o total de atributos. Para a realização desse experimento escolhemos como modo de testes dos resultados a técnica estatística de validação cruzada com configuração de 3 dobras e 2 validações. Para a visualização dos resultados foram selecionadas as seguintes características: taxa(%) de classificação correta e tempos(s) de treinamento e classificação. Dessa forma foi possível analisar o desempenho do classificador à medida que os atributos são reduzidos utilizando Análise Discriminante de Fisher.

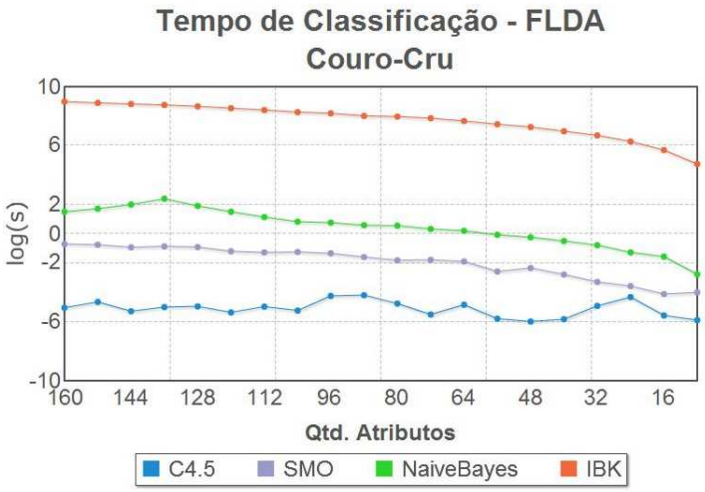
O tempo total para a realização dos 20 experimentos foi de 16 horas para C4.5 e 21 horas para IBK, 18 horas para SMO e 12 horas para NaiveBayes. Os Gráficos 6.5(a), 6.5(b) e 6.5(c), ilustram os resultados com base na porcentagem de acerto e os tempos de treinamento e classificação dos algoritmos C4.5, SMO, IBk, e NaiveBayes.



(*a*)



(*b*)



(*c*)

Figura 6.5: Comportamento do classificador, (a) classificação correta(%), (b) tempo de treinamento em $log_2(s)$, (c) tempo de classificação em $log_2(s)$.

6.7.1 Resultados e Análise

Com base nos resultados, podemos observar que a taxa de acerto dos classificadores tiveram um bom desempenho. O melhor caso foi a técnica IBk com 73.48% e o pior caso $NaiveBayes$ com 23.42%. Também acreditamos que as grandes variações nos defeitos marcados do couro cru e a baixa quantidade de imagens utilizadas influenciaram nos resultados da redução.

Analisando o comportamento do gráfico 6.5(a) podemos verificar que mesmo com a redução de atributos, mas com pequenas variações da taxa de acerto, a Análise Discriminante de Fisher apresentou um bom desempenho mantendo o nível dos classificadores quase constante. Pelo gráfico também pôde ser verificado que a porcentagem de acerto se mantém constante até um certo nível, e à medida que a quantidade de atributos é decrementada, a taxa de acerto diminui. Para o algoritmo C4.5 a estabilidade é encontrada a partir de 128 atributos com 69.53% de acerto, SMO com 136 atributos e taxa 36.39% de acerto, IBk com 104 atributos e taxa 71.10% de acerto e NaiveBayes com 152 atributos e taxa 36.95% de acerto.

Podemos observar também que para alguns algoritmos de classificação o comportamento é o mesmo quanto a classificação sem a redução de atributos. Novamente a técnica de IBk se destacou e a técnica NaiveBayes apresentou os piores resultados. Mas se avaliarmos a taxa de aprendizagem com o custo de aprendizagem e classificação, a técnica que mais se destacou foi C4.5, pois seu resultado é o que mais se aproxima da classificação de IBk e seu tempo de classificação é muito menor do que as outras técnicas.

Para entendermos melhor, se capturarmos o tempo de classificação referente a melhor taxa de classificação para a técnica de C4.5 com valor 0.0465ms e IBk com valor 278.392ms temos que, IBk leva um tempo aproximadamente 7.000 vezes maior do que C4.5, para conseguir um resultado semelhante de classificação. A Tabela 6.8 ilustra uma breve comparação da técnica C4.5 sobre as outras técnicas utilizadas levando em consideração o tempo de classificação, a partir da menor e maior quantidade de atributos reduzida.

Tabela 6.8: Quantidade de Atributos Vs. Tempo de Classificação

Qtd. Atributos	C4.5(ms)	IBk(ms)	SMO(ms)	NaiveBayes(ms)
8	0.036	27.947	0.044	0.225
152	0.049	479.105	0.4548	4.600

6.8 Redução de atributos sobre imagens com defeitos no Wet-Blue utilizando a técnica tradicional de Análise Discriminante de Fisher

O objetivo da realização desse segundo experimento foi analisar o desempenho da classificação de defeitos do couro bovino no estágio Wet-Blue, quando a quantidade de atributos é reduzida. Esse tipo de experimento junto com o experimento anterior irá nos ajudar, a verificar o poder da redução de atributos quanto a sua capacidade de redução de informações sem que haja perda na qualidade na discriminação dos defeitos e conseqüentemente diminuindo o tempo de aprendizagem e classificação. Para bases de dados maiores, onde a velocidade de resposta é importante, essas características acabam se tornando de suma importância.

Para a realização desse experimento utilizamos somente as imagens do couro na fase de curtimento chamada Wet-Blue. Para mantermos o padrão de experimentos utilizamos as mesmas técnicas de extração de atributos, aprendizagem e configuração dos testes utilizando validação cruzada, que o experimento com a redução de atributos sobre imagens do couro cru utilizaram.

Dessa forma, a partir do número total de atributos, 160, e junto à técnica de redução de atributos por Fisher, foram geradas as 20 bases de aprendizagem reduzidas. O tempo total para a realização dos 20 experimentos foi de 17 horas para C4.5, 19 horas para SMO, 24 horas para IBk e 15 horas para NaiveBayes. Os Gráficos 6.6 (a), 6.6(b) e 6.6(c) ilustram os resultados com base na porcentagem de acerto e os tempos de treinamento e classificação.

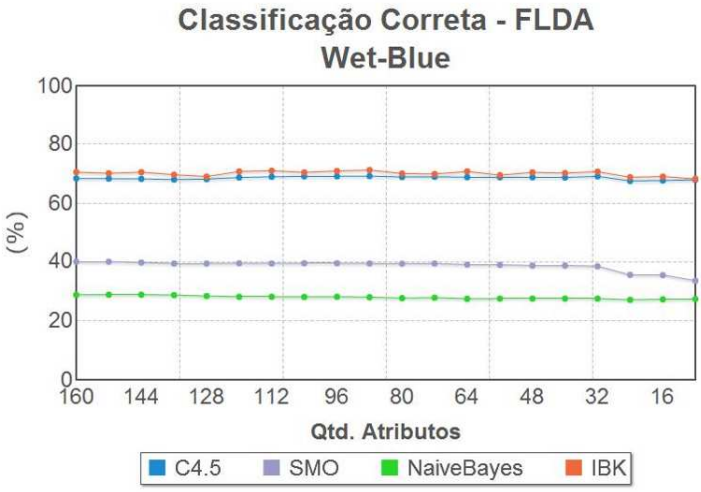
6.8.1 Resultados e Análise

Em experimentos de trabalhos anteriores com base de amostras mais simples [25] e [2], conseguimos verificar que a redução de atributos trouxe um grande impacto em termos de desempenho na classificação e tempos de aprendizagem. Utilizando 66 atributos baseados em matriz de co-ocorrência, mapas de interação e HSB e uma quantidade de 7.680 amostras sendo eles: 1.368 para carrapato, 411 para risco, 863 para sarna, 778 marca-ferro, 3.218 fundo e 1.042 para sem defeito conseguiu-se chegar no melhor caso, em um resultado de 100% de acerto e pior caso 84%. Mas isso ocorreu pelo importante fato de que foram utilizadas somente quatro imagens para o experimento.

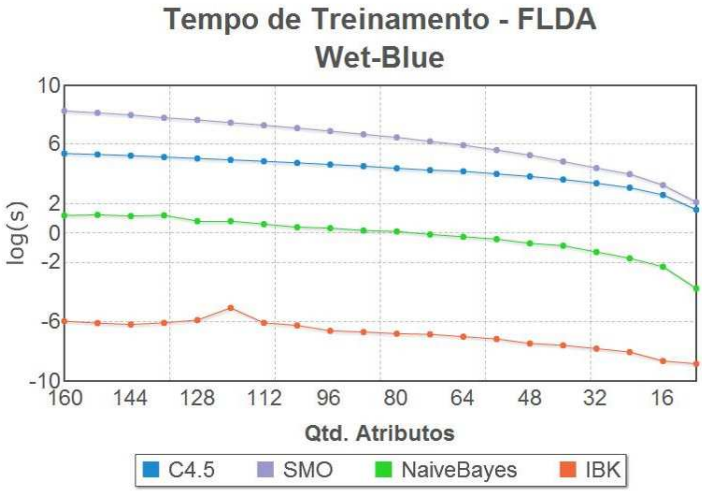
Para esse experimento, aumentamos sua complexidade e agora estamos utilizando no total 50 imagens selecionadas sobre um conjunto de 150. Sobre esse conjunto de imagens são encontrados também defeitos, como: carrapato, berne, risco, esfola e furo com diferentes formas visuais para sua classificação, mas também grandes semelhanças entre cada tipo de defeito. Um exemplo para esse caso é o conjunto de imagens da Figura 6.7, em que temos 3 amostras diferentes representando cada tipo de defeito, mas semelhantes a outros tipos, como: carrapato com berne, berne com furo e risco com esfola.

Para facilitar a visualização dos resultados foram criados os gráficos 6.6(a) e 6.6(b) e 6.6(c) com o objetivo de mostrar que a redução de atributos é uma etapa muito importante para classificação de imagens. À medida que a quantidade de atributos é decrementada, o tempo de treinamento e classificação diminuem mostrando assim que a utilização de um técnica para redução de atributos, como análise discriminante de Fisher pode dar eficiência, velocidade e confiança para um classificador automático.

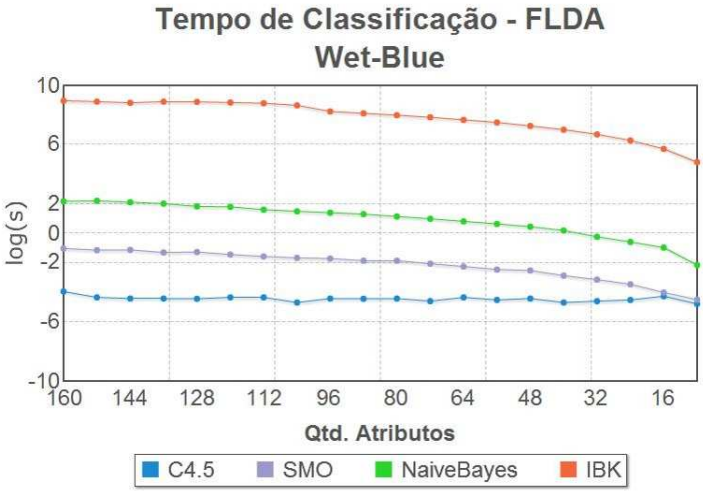
Analisando os resultados, podemos observar que a redução apresentou novamente bons efeitos chegando a pontos onde existe a estabilidade dos resultados de classificação. O melhor caso foi novamente obtido pela técnica IBK com 71.89% de acerto e pior caso para a técnica $NaiveBayes$ com 27.22%. A estabilidade é encontrada para IBK em 136 atributos com 69.48% de acerto, C4.5 com estabilidade em 112 atributos e 69.02%, SMO com 152 atributos e 40.22% e $NaiveBayes$ com 88 atributos e 28.09% de acerto. Podemos analisar também que mesmo com uma base de dados mais complexa ou mais realista ao problema, mostramos que a redução de atributos nos permitiu um ganho importante. Para o problema de classificação de defeitos, em que existe uma complexidade na discriminação, poderemos utilizar a técnica de LDA para nos garantir um maior desempenho tanto na taxa de acerto quanto no tempo de aprendizagem na classificação automática.



(*a*)



(*b*)



(*c*)

Figura 6.6: Comportamento do classificador, (a) classificação correta(%), (b) tempo de treinamento em $log_2(s)$, (c) tempo de classificação em $log_2(s)$.

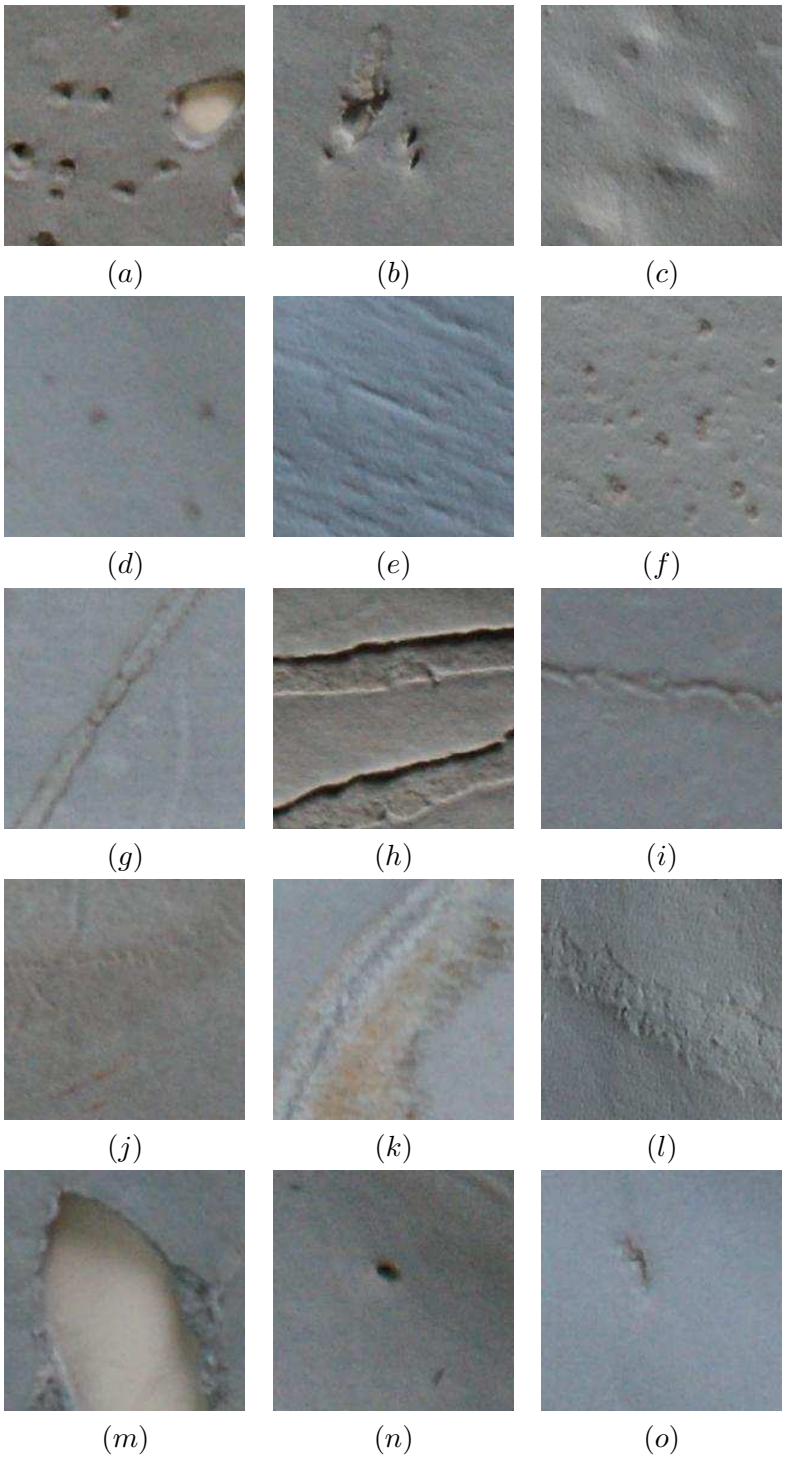


Figura 6.7: Amostras visualmente semelhantes (a), (b), (c) defeito berne, (d), (e), (f) defeito carrapato, (g), (h), (i) defeito risco, (j), (k), (l) defeito esfola e (m), (n), (o) defeito furo.

6.9 Experimento de simulação para o Problema de Singularidade

Os experimentos até aqui realizados foram feitos com objetivo de nos ajudar a termos uma base e análise experimental, sobre o desempenho na classificação de imagens do couro bovino tanto em peça do couro cru quanto Wet-Blue e análise sobre o desempenho da redução de atributos utilizando a técnica tradicional LDA.

A partir desses experimentos conseguimos analisar que a redução de atributos utilizando a análise discriminante de Fisher se comportou de maneira satisfatória, mostrando que para alguns casos existe a estabilidade da melhor taxa de classificação correta. Mas até o momento as bases de dados criados para a realização dos experimentos, foram bases com menos chances a problemas, com imagens e atributos significantes.

O problema de classificação automática de padrões em couro bovino pode ser comparado com o problema para reconhecimento de faces, onde existe uma quantidade alta de atributos e uma grande dificuldade na captura de imagens para amostras. O problema de baixa quantidade de imagens de couro bovino acontece pelo fato de existir uma grande dificuldade na captura e rastreamento dessas imagens saindo do abate e sendo acompanhada até a fase de curtimento Wet-Blue. Outro problema também se encontra no difícil acesso a ambiente para captura em frigoríficos, em que muitos não possuem uma iluminação controlada e a maioria dos couros é trabalhada no próprio chão, que se encontra sempre molhado dificultando assim uma padronização de ilumuninação sobre os mesmos.

Para a realização desses experimentos criamos uma base diferente das bases de dados até o momento criadas em que temos uma quantidade de amostras e atributos significativamente satisfatórios. Iremos realizar essa mudança por que poderão existir casos, em que a quantidade de amostras não poderá ser tão grande e sim discriminatório suficiente, para que diminua o custo de aprendizagem e classificação.

Os trabalhos correlatos [38] e [9] mostram que para problemas onde uma quantidade de amostras é muito inferior do que a quantidade de atributos para a discriminação das classes de classificação, poderá ocorrer problemas na matriz de dispersão intra-classes quando se utilizado a técnica tradicional LDA. Para isso então simulamos uma situação como essa, para sabermos se realmente isso poderá ser um problema na redução de atributos.

Para criarmos essa base de amostras utilizamos a mesma base de imagens dos primeiros experimentos, tanto para couro cru e Wet-Blue e mudaremos as configurações dos métodos de extração de atributos que utilizamos, para aumentar ainda mais os atributos extraídos. Para a geração das novas amostras foi utilizada a mesma funcionalidade de geração aleatória uniforme, mas gerando agora uma quantidade de 100 amostras para cada tipo de defeito, totalizando assim 800 amostras para o couro cru e 800 para Wet-Blue.

Seguindo a linha dos experimentos anteriores, iremos utilizar o módulo de geração de experimentos para a extração de atributos, mas agora as propriedades de cada técnica de extração serão diferentes. Essas mudanças foram: matriz de co-ocorrência com variações de (0^o a 360^o com intervalo de 1^o e distância de 1 pixel) extraindo 2.520 atributos, mapas de interação com variações de (0^o a 180^o com intervalo de 10^o e distância inicial de 0 a 2 com intervalo de 1 pixel) extraindo 7 atributos, Filtros de Gabor extraindo 15 atributos, valores médio do histograma de cada componente de cor de H(matiz), S(saturação) e B(brilho) com 6 atributos, mais 6 atributos dos valores discretizados em 3 intervalos de (HSB) e (RGB), e além

disso iremos considerar cada pixel da amostra como atributo, sendo que cada amostra possui tamanho de 40x40 pixel, resultando assim em 1.600 atributos. A Tabela 6.9 mostra com mais detalhes os atributos extraídos.

Tabela 6.9: Atributos extraídos segundo experimento

Método de Extração	**Quantidade de Atributos**
Média (H), (S), e (B)	3
Média (R), (G) e (B)	3
HSB (Discretizado em 3 intervalos)	3
RGB (Discretizado em 3 intervalos)	3
Mapas de Interação	7
Matriz de Co-ocorrência	2.520
Filtros de Gabor	15
Píxels das amostras	1.600
Total	**4.154**

6.9.1 Resultados e Análise

A partir da base de aprendizagem gerada, realizamos a tentativa na redução de dimensão utilizando a técnica de Fisher tanto na base couro-cru, quanto na base Wet-Blue e como trabalhos correlatos [5] e [17] apresentaram, ocorreu o problema de matriz singular. Analisando esse problema, temos que a quantidade de atributos gerada é de 4.154 e que para isso trabalhos como [34], mostram que teríamos pelo menos, para calcular a matriz de dispersão de uma matriz de tamanho 4.154 * 4.154, aproximadamente 18M. Acreditamos assim que o problema além de ser influenciado pela difícil computação de grandes matrizes, mas também possa ser influenciado pela baixa quantidade de amostras, tendo pelo menos 18M de amostras, evitando assim, a sua não-degeneração da matriz de dispersão intra-classe.

Para o nosso problema de classificação de tipos de defeitos em couro bovino, o aumento da quantidade de amostras é muito difícil, pelo seguinte fato: conseguimos até o momento imagens significativamente consideráveis em termos de classificação, mas para aumentarmos o número dessas imagens é preciso a realização de mais viagens levando em consideração seu custo, e principalmente a dificuldade na padronização em sua captura. Visto que para curtumes e frigoríficos diferentes temos também ambientes e iluminações diferentes.

Alguns trabalhos tentam resolver esse problema de baixa quantidade de amostras reduzindo a quantidade de atributos ou removendo os atributos com baixa variância, tentando evitar assim, a não singularidade da matriz de dispersão intra-classe, mas conseqüentemente, esses tipos de soluções, deixam as bases de aprendizagem pobremente estimadas. Para evitar soluções como essas, pesquisamos e implementamos outras técnicas baseadas em LDA, que realizam o processo de redução de atributos com o principal objetivo de evitar casos como esse, que ocorrem o problema de singularidade. A seguir são apresentados os experimentos com as técnicas pesquisadas.

6.10 Redução de atributos baseadas em LDA, utilizando imagens com defeitos sobre o Couro-Cru e Wet-Blue.

A partir dos experimentos apresentados conseguimos verificar que a redução de atributos possui um comportamento satisfatório em termos de melhoria no tempo de aprendizagem e classificação sem que haja grande perda na discriminalidade entre os padrões das classes. Verificamos também que para bases de aprendizagem que possuem um conjunto pequeno de amostras e uma quantidade alta de atributos, ocorreu o problema de singularidade na redução de atributos utilizando a técnica tradicional de LDA.

A partir disso, foram pesquisadas e implementadas técnicas baseadas na técnica de LDA, que possuem variações em seu comportamento que fazem com que problemas como a singularidade sejam tratados da melhor forma sem que se perca informações importantes para a redução de atributos.

Para demonstrarmos a eficiência de cada uma dessas técnicas de redução de atributos realizamos uma análise e comparação aplicada em bases de aprendizagem, contendo informações com relação a imagens de defeitos do couro bovino cru e imagens do couro bovino em fase de curtimento Wet-Blue.

A maioria das técnicas apresentadas defende em seus trabalhos sua capacidade de redução de informações, tentando manter a sua discriminalidade, sem perder informações importantes dos conjuntos de amostras. Dessa forma para termos uma boa base de avaliação das técnicas, utilizamos como métrica para avaliação, a taxa de classificação correta comparada a cada algoritmo de aprendizagem aqui trabalhados: C4.5, IBk, SMO e NaiveBayes.

Nos experimentos a seguir, iremos mostrar uma comparação das técnicas de redução de atributos que apresentamos como soluções para problemas que envolvem a singularidade. Para isso os experimentos foram divididos da seguinte forma: Experimentos sobre bases de amostras que ocorreram e não a singularidade em couro cru e Wet-Blue. Isso irá nos ajudar a mostrar o desempenho de cada técnica quanto sua eficiência sobre a técnica tradicional, e eficiência quanto à redução sobre problemas que envolvem a técnica de LDA. As técnicas apresentadas para redução de atributos baseada na técnica tradicional LDA são: DLDA, YLDA, FisherFace, CLDA e KLDA.

6.10.1 Experimento sobre as bases couro-cru e wet-blue sem problema de singularidade

Para esse experimento utilizamos a base de amostras referente aos primeiros experimentos sobre o couro-cru e wet-blue, em que foram utilizadas 20 imagens de couro cru e 50 imagens para couro curtido Wet-Blue, correspondendo à 2.000 amostras para cada defeito. Os atributos dessas bases de aprendizagem são relacionadas às técnicas de matriz de co-ocorrência, mapas de iteração, HSB e filtros de Gabor, totalizando assim 160 atributos.

Da mesma forma que o experimento sobre a técnica tradicional LDA, pegamos o total de atributos, sendo 160, e criamos 20 bases de aprendizagem para cada técnica, cada uma contendo um conjunto de a_i atributos, em que i varia do decremento de 5% sobre o total de atributos.

A característica que utilizamos para comparação foi somente a taxa de classificação correta e os algoritmos de classificação foram os mesmos que os experimentos anteriores, que são: C4.5,

IBK, SMO e NaiveBayes. Os gráficos 6.8 e 6.9 ilustram os resultados da redução de atributos utilizando as técnicas propostas sobre casos em que não ocorrem a singularidade.

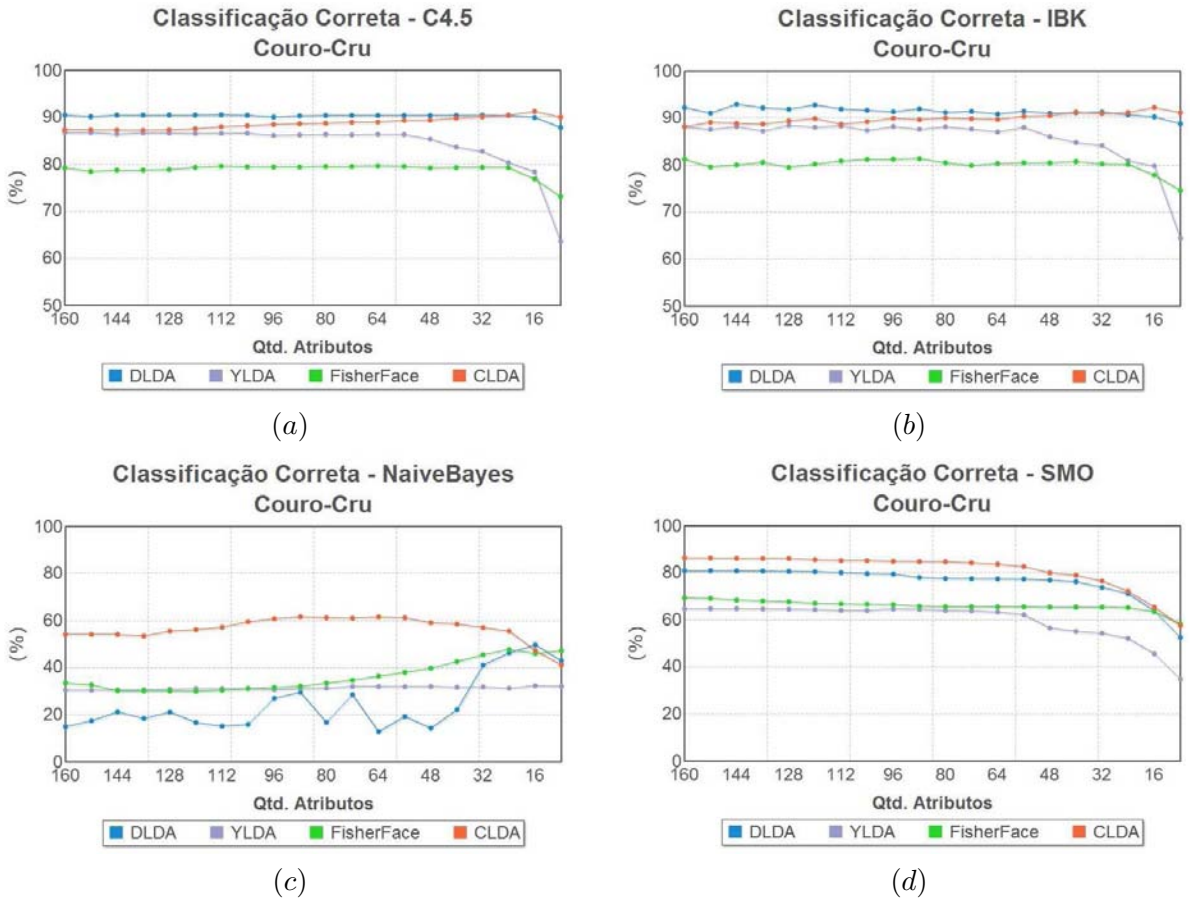


Figura 6.8: Comportamento e Comparação de técnicas de Redução de Atributos para bases de aprendizagem que não ocorrem a singularidade no Couro Cru.

6.10.2 Experimento sobre a base couro-cru e wet-blue com problema de singularidade

Para esse segundo experimento comparativo das técnicas de redução de atributos, utilizamos a base de aprendizagem gerada que focou na simulação do problema de singularidade. Essa base de aprendizagem contém 100 amostras para cada tipo de defeito, totalizando 800 amostras para couro cru e 800 amostras para wet-blue.

De forma diferente dos primeiros experimentos de redução de atributos, avaliamos esse experimento gerando apenas uma base de aprendizagem, contendo a quantidade de 2.077, referente à metade do número total de atributos. O motivo por essa mudança está em que nesse experimento, queremos somente apresentar quais as técnicas e seus respectivos desempenhos quanto à redução de informações, em casos em que a singularidade.

Os atributos gerados para essa base de aprendizagem foram no total de 4.154, sendo eles referentes as técnicas matriz de co-ocorrência, mapas de iteração, HSB e filtros de Gabor. E

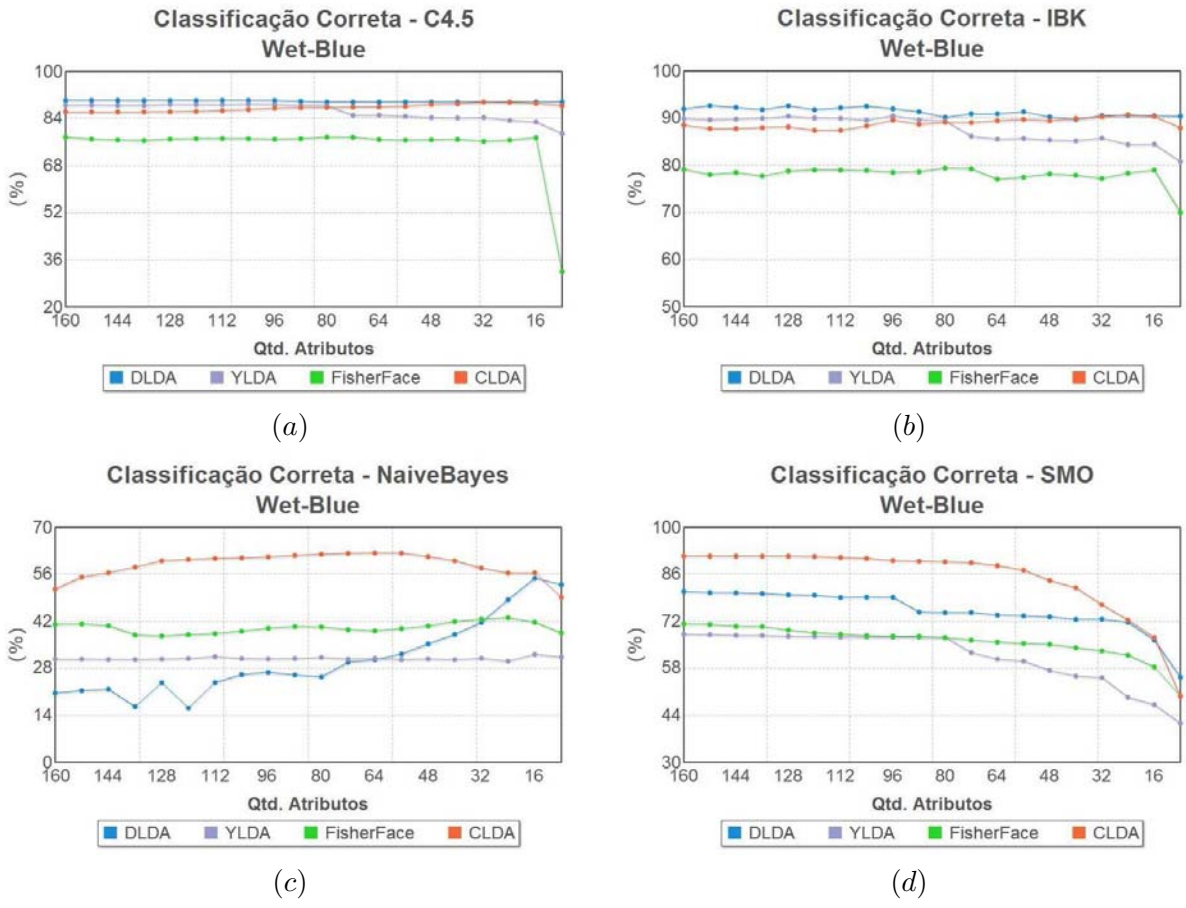


Figura 6.9: Comportamento e Comparação de técnicas de Redução de Atributos para bases de aprendizagem que não ocorrem a singularidade no Couro Wet-Blue.

como também no experimento anterior, utilizamos a taxa de classificação correta como meio de comparar o desempenho de cada técnica aqui apresentada. A Tabela 6.10 e 6.11 ilustram os resultados de redução de atributos, para as bases de imagens do couro cru e wet-blue, em casos que ocorre a singularidade.

Tabela 6.10: Resultado classificação(%) couro Wet-Blue

Classificador	**CLDA**	**FisherFace**	**YLDA**	**DLDA**	**KLDA**
C4.5	85.18	75.96	81.48	88.45	72.24
IBk	90.3	78.21	83.18	88.32	74.21
Naives Bayes	50.14	30.27	39.66	44.74	20.12
SMO	78.11	55.24	67.53	79.88	48.90

Tabela 6.11: Resultado classificação(%) Couro Cru

Classificador	**CLDA**	**FisherFace**	**YLDA**	**DLDA**	**KLDA**
C4.5	87.91	77.75	84.22	88.98	72.78
IBk	92.23	80.5	85.02	90.08	76.73
Naives Bayes	52.75	31.09	42.16	46.85	25.43
SMO	80.02	66.56	69.48	81.76	54.16

6.10.3 Análise dos Resultados

Com bases nesses experimentos conseguimos novamente mostrar o quanto é importante o uso de técnicas na redução de informações em aplicações onde seu principal objetivo está na confiabilidade do classificador e baixo custo de processamento na aprendizagem e classificação. Os experimentos também mostraram que mesmo em situações em que ocorrem problemas de singularidade é possível reduzir a quantidade de informações e ao mesmo tempo manter a discriminalidade entre as classes de classificação.

Durante a realização desse experimento ocorreu um problema no uso da técnica de Kernel(KLDA) aplicada sobre a base de aprendizagem sem problema de singularidade. O problema foi o elevado tempo de processamento na redução dos atributos, não conseguindo assim chegar a nenhum resultado, após 48horas em execução. O motivo disso foi que a técnica de kernel realiza projeções dos valores dos atributos no espaço de kernel, criando assim novos valores que são calculados levando em consideração a quantidade total de amostras. Dessa forma pela grande quantidade de atributos e amostras, ficou inviável seu cálculo para essa base. Mas para o problema que ocorre a singularidade onde a base de amostras equivale a aproximadamente 3.85% do tamanho da base que não ocorre a singularidade, não houve o problema no uso da técnica KLDA.

Analisado os resultados para o problema que não apresenta a singularidade, identificamos um grande desempenho das técnicas de redução, apresentando até mesmo resultados consideráveis e mais elevados do que a própria técnica tradicional de análise discriminante de Fisher. Para os dois problemas de classificação couro-cru e wet-blue foram obtidos resultados próximos de classificação e grande semelhança de desempenho à medida que a quantidade de informações era decrementada.

Para o experimento sobre casos com singularidade em que redução foi realizada uma única vez, com um valor da metade do total de atributos, conseguiu-se identificar o quanto as técnicas se mostraram eficientes, pois mesmo diminuindo 50% do total de informações é possível bons resultados de classificação. No caso da base de aprendizagem sem problemas de singularidade, conseguimos para todas as técnicas desempenhos semelhantes e mais elevados, comparados ao uso da técnica de LDA.

As técnicas que mais se destacaram na redução de atributos para os dois formatos de experimentos que realizamos, em que ocorrem ou não a singularidade, foram as técnicas CLDA e DLDA. Para o couro wet-blue sem singularidade, os melhores casos foram utilizando 24 atributos com taxa de 91.47% de acerto para a técnica CLDA, e 144 atributos com taxa de 92.33% de acerto para a técnica DLDA. Para o couro cru sem singularidade, os casos que se destacaram foram utilizando 16 atributos para CLDA com 92.28% de acerto e 136 atributos com taxa de 93.32% de acerto utilizando a técnica DLDA. Para o couro cru e wet-blue em casos que ocorrem a singularidade, novamente as técnicas CLDA e DLDA se mostraram eficientes chegando a 92.23% e 90.08% para o couro cru e 90.3% e 88.32% respectivamente para o couro wet-blue. Mas também podemos observar que a técnica YLDA teve um bom desempenho na redução, chegando ao valor em seus melhores resultados de 83.18% para Wet-Blue e 85.02% para couro cru em casos que ocorrem a singularidade e 90.75% para Wet-Blue e 88.5% para couro cru em casos que não ocorrem a singularidade, diferente das técnicas FisherFace e KLDA, que apresentaram os piores resultados na maioria dos casos. Para a classificação das bases de aprendizagem reduzidas, novamente o algoritmo IBk apresentou os melhores resultados, mostrando-se assim de grande interesse de estudo e uso. Mas apesar de ter os melhores resultados, IBk acaba sendo ineficiente quando tratado sobre tempo de aprendizagem e classificação, como ilustrado nos primeiros experimentos. Diferente para o algoritmo C4.5, com tempo de aprendizagem e classificação muito menor do que IBk, mas com poder de classificação satisfatório.

Analisando melhor os piores casos, temos que as técnicas FisherFaces e KLDA, apresentaram os piores resultados. Para problemas de singularidade, foi conseguido um resultado no melhor caso para FisherFace de 78.21% para couro Wet-Blue e 80.5% para couro cru e KLDA com 76.73% para couro cru e 74.21% em wet-blue. No caso de problemas que não ocorrem a singularidade a pior taxa de acerto também foi da técnica FisherFace, classificando sobre o couro Wet-Blue, com um resultado de 32% de acerto. Analisando os algoritmos de aprendizagem, novamente podemos observar que o algoritmo que se mostrou menos eficiente foi NaiveBayes, que na maior parte das bases de aprendizagem reduzidas, obteve um desempenho inferior.

Para entendermos melhor os resultados obtidos podemos realizar a seguinte análise. A técnica CLDA, tende a utilizar o espaço nulo da matriz de dispersão intra-classes, que é a base do problema de singularidade, para maximizar a matriz de dispersão inter-classes e assim encontrar as informações mais discriminantes sobre o espaço de atributos. Seguindo o mesmo conceito, a técnica DLDA, concentra-se também na análise das matrizes inter-classe e intra-classe, através da diagonalização de suas matrizes simétricas, descartando assim o espaço nulo da matriz de dispersão inter-classe.

Um ponto importante que podemos observar é que nenhuma dessas duas técnicas consideradas pelos nossos experimentos como os melhores casos, descartam as informações do espaço nulo da matriz de dispersão intra-classe, o que em muitos trabalhos citam como o principal causador de sua singularidade e instabilidade.

Diferentemente das técnicas FisherFace e YLDA, o centro de sua funcionalidade é justamente utilizar em sua execução, a remoção das informações menos discriminantes sobre a matriz intra-

classe, utilizando a técnica de análise componentes principais. Mas acreditamos também, como o trabalho, [39], que a remoção desse conjunto de informações ou de qualquer outro, tende a influenciar diretamente no desempenho e diminuir a discriminalidade entre as classes de classificação. No caso da técnica KLDA, a baixa taxa de acerto pode estar relacionado a sua função em encontrar as informações mais discriminantes, não tratando assim diretamente casos que influencie no problema de singularidade. Mas alguns trabalhos [40] e [18] mostram que a mudança no uso de kernel aplicada sobre soluções que tratam problemas da instabilidade da matriz intra-classe podem trazer ganhos mais satisfatórios.

6.11 Classificação automática de defeitos em comparação à classificação por especialistas na área.

Esta seção irá apresentar resultados visuais na detecção automática de defeitos no couro bovino utilizando o módulo de classificação automática do DTCOURO. Como método de avaliação do nosso trabalho aqui apresentado iremos realizar como fase final, em nossos experimentos, uma comparação na classificação automática de defeitos aqui apresentados com um especialista treinado na área. O objetivo disso será uma avaliação da base de dados criada e o classificador automático, com base na classificação realizada por um especialista que diariamente realiza a identificação e reconhecimento de diversos tipos de defeitos.

O resultado desse experimento irá nos ajudar a identificar os principais pontos que precisam ser melhorados e acrescentados para evolução do projeto. Também iremos conseguir identificar quais tipos de defeitos que as técnicas de extração e treinamento possuem mais dificuldade de classificação, para assim focarmos especificamente em problemas mais isolados dando mais garantia de acerto e desempenho.

Com uma comparação entre o sistema de classificação automática e um especialista, poderemos ter uma resposta interessante em termos de credibilidade. Isso significa que quanto mais próximo da classificação por um ser humano, mas perto estaremos de um sistema comercialmente utilizável. Mas sabemos também que sua constante melhoria é de grande importância.

Para a realização desse experimento utilizamos todos os módulos e técnicas implementadas. Selecionamos a base de aprendizagem com a melhor taxa de classificação e realizamos todos os passos necessários que um sistema de visão computacional precisa para uma inspeção automática.

Para todo esse processo acontecer utilizamos o módulo de classificação automática. Esse módulo ficou responsável em realizar a comunicação entre os módulos de aquisição de imagem, geração de experimentos, redução de atributos e regras de classificação, realizando todos os processos desde a extração de atributos passando para a fase de criação da base de aprendizagem, seguida da redução do espaço de atributos para a geração da nova base reduzida e por fim o treinamento e varredura sobre a imagem para a classificação. A próxima seção descreve o módulo de classificação automática.

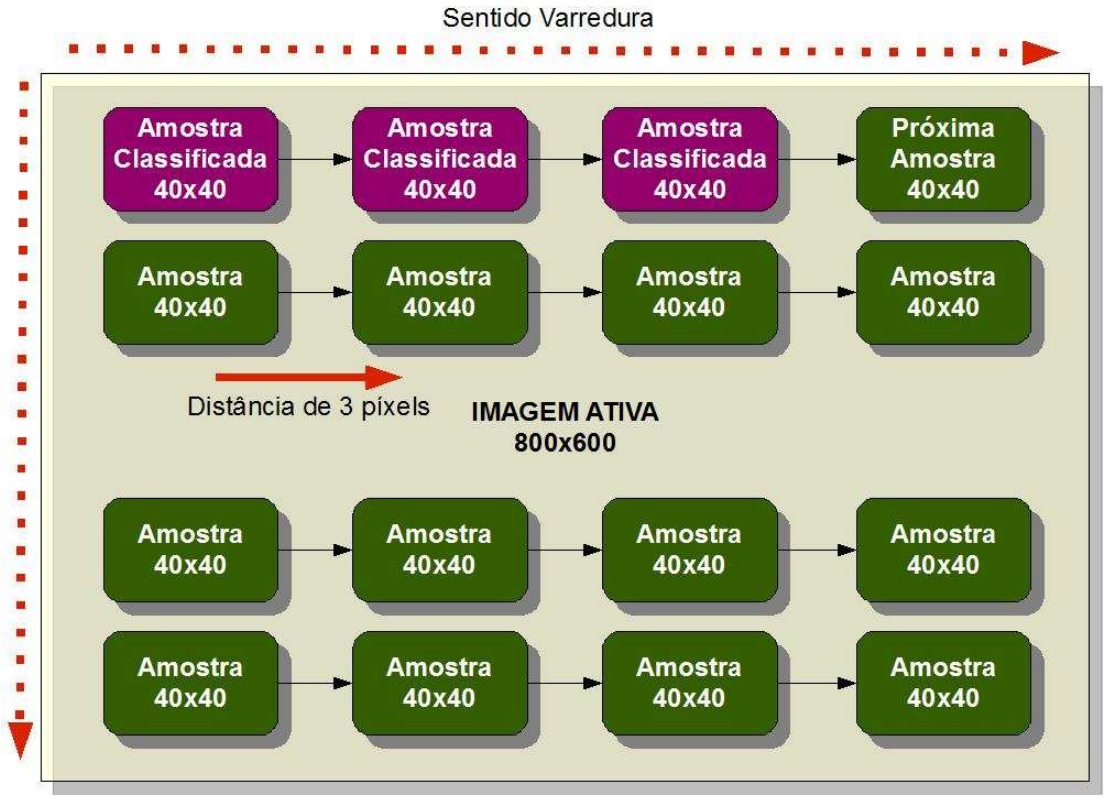
6.11.1 Módulo DTCOURO para classificação automática

O módulo aqui utilizado para classificação automática das imagens de couros bovinos foi implementado, com o principal objetivo de proporcionar uma avaliação visual dos resultados de

detecção e classificação sobre uma imagem presente ou não no conjunto de treinamento.

Para a realização de qualquer experimento utilizando esse módulo, alguns itens precisam ser passados inicialmente, como os parâmetros: imagem ativa, base de aprendizagem, algoritmo de treinamento e classificação, métodos de extração de atributos, seleção de cores para pintar o defeito ao ser encontrado, dimensão das amostras capturadas e intervalo em pixels de cada amostra para varredura.

Após a configuração desses itens, o processo de classificação automática acontece da seguinte forma. O sistema varre a imagem selecionada ativa e percorre capturando desde o inicio da imagem, amostras em um intervalo de pixels selecionado realizando a extração e redução de atributos. Com esse vetor de atributos capturado da amostra, o módulo realiza então a classificação. Essa varredura é feita desde o canto superior esquerdo da imagem até o canto inferior direito, garantindo assim a captura e classificação da imagem por inteiro. A Figura 6.10 ilustra de modo gráfico esse funcionamento.



(a)

Figura 6.10: Modelo de varredura para a classificação automática.

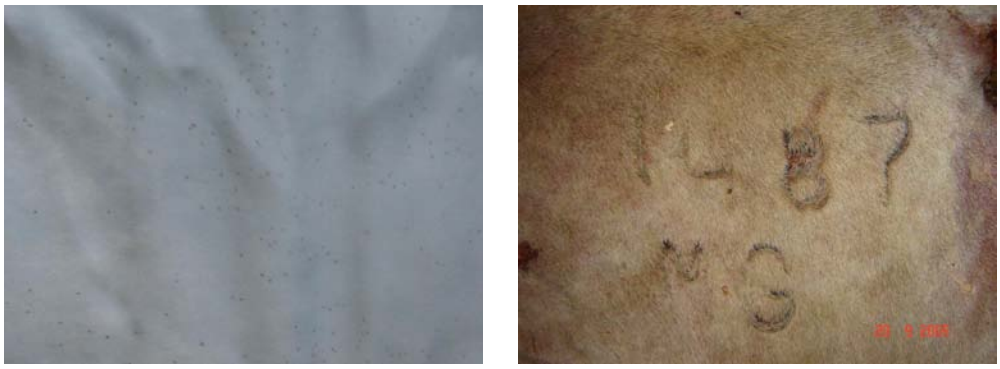
6.11.2 Configuração do experimento

As bases de aprendizagem utilizadas são as mesmas que para o primeiro e segundo experimento para classificação de imagens do couro cru e wet-blue, quando foram utilizadas 2.000 amostras para cada tipo de defeito totalizando 16.000 para couro cru e wet-blue. Sobre essa base de aprendizagem contendo 160 atributos geramos utilizando a técnica de DLDA, uma

nova base reduzida com a quantidade de 136 atributos. Escolhemos essa quantidade e essa técnica, pelo fato que no experimento de redução foram os que obtiveram os melhores resultados com uma taxa de acerto de 91.47% para wet-blue e com o mesmo tamanho de atributos obteve 93.32% para couro cru.

Como algoritmo de classificação foi escolhido o método C4.5, pelo motivo que na maioria dos experimentos aqui apresentados, foi o que teve um dos melhores resultados na classificação. A dimensão das amostras para varredura foi de 40x40 pixels, sendo a mesma utilizada para a geração de amostras na criação da base de aprendizagem.

Duas imagens que não pertencem a base de aprendizagem foram utilizadas para a classificação automática visual, sendo uma do couro cru e outra de couro wet-blue. Essas imagens apresentam diversos tipos defeitos, que variam entre fácil identificação até um nível mais alto para a classificação. Essas imagens também foram enviadas para nossa especialista, para que fosse feita assim uma identificação e classificação dos defeitos, para em seguida realizaremos uma breve comparação do desempenho do nosso sistema para classificação. A especialista selecionada faz parte do projeto DTCOURO. As duas imagens utilizadas seguem na Figura 6.11.



(a) (b)

Figura 6.11: Imagens utilizadas para testes utilizando o módulo de classificação automática do DTCOURO. (a) couro wet-blue (b) couro cru.

O módulo implementado de classificação automática visual, também nos possibilita a especificação do tamanho da amostra a ser classificada. Isso permite uma avaliação quanto ao tempo de processamento para classificação da imagem, com relação à qualidade na classificação. Para isso então iremos realizar para cada imagem, dois tamanhos diferentes de amostras de classificação, sendo elas de tamanhos 40 e 5 pixel. Quando informado a opção de 5 pixel, a classificação será realizada na escala original 40x40, mas o pixel a ser colorido com a classe referente, serão os pixels do centro até os quatro cantos da amostra, garantindo uma precisão maior na visualização do defeito classificado. Para termos uma melhor identificação da classificação das amostras, cada tipo de defeito foi colorido com uma cor diferente, dando assim um maior destaque na região classificada. A Figura 6.12 ilustra a lista de cores com seus respectivos defeitos.

Figura 6.12: Legenda de cores para a classificação automática.

6.11.3 Resultado Classificação Sistema Vs. Especialista

O módulo para classificação automática gera como resultado de cada imagem, uma cópia, contendo as marcações nas regiões que foram identificados os defeitos. As Figuras 6.13 e 6.14 mostram as imagens classificadas pelo sistema em escala de 5x5 e 40x40 pixels divididas em couro cru e couro curtido wet-blue, seguida também das classificações manuais.

6.11.4 Análise dos resultados

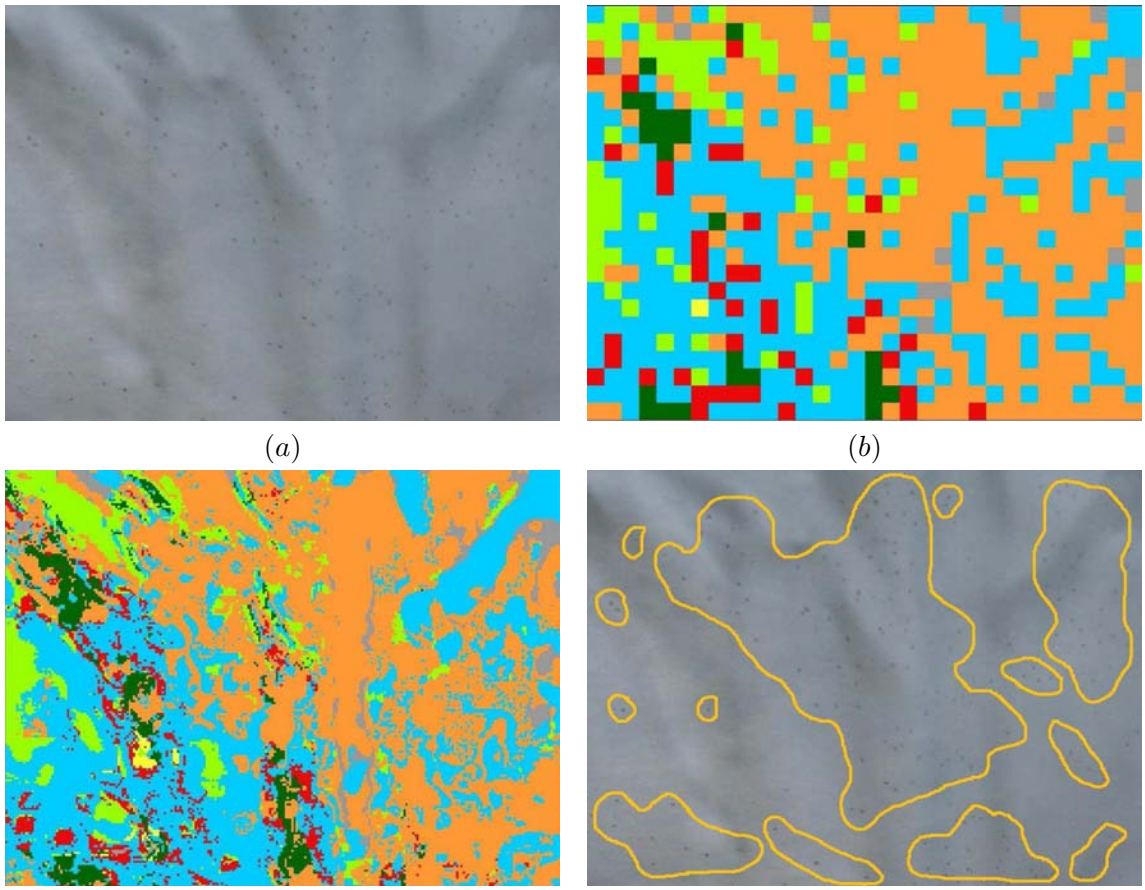
Como pode ser observado, as imagens classificadas automaticamente pelo sistema apresentaram regiões que não deveriam ser marcadas, ou que foram classificadas defeitos de maneira incorreta, isso comparado com as imagens marcadas manualmente pelo especialista, como ilustrado nas Figuras 6.13 e 6.14.

Isso significa que para esse experimento foram apresentadas diversas classificações "falso positivo", isto é, o classificador identificou como um tipo de defeito, acreditando que seria a mais correta. Isso ocorre por diversos motivos, mas entre eles se destaca a quantidade e qualidade de amostras utilizadas, sendo não significativamente suficiente para discriminar todos os defeitos.

Podemos observar também que o parâmetro de escala para classificação, ajudou à manter uma qualidade de visualização sobre as amostras classificadas. Na primeira imagem sobre o couro curtido wet-blue podemos observar que o couro está com uma quantidade alta de carrapato e que mesmo sendo difícil a sua identificação, o sistema automático conseguiu ainda classificar a maior parte da área, chegando muito perto ou mais preciso do que a classificação pelo especialista.

No caso do couro cru, podemos observar que toda a região de marca-fogo foi identificada pelo sistema e também pelo especialista, mas apresentando uma característica muito interessante qual foi novamente a precisão do classificador automático, em regiões de contorno sobre a marca de fogo classificando assim com uma esfola.

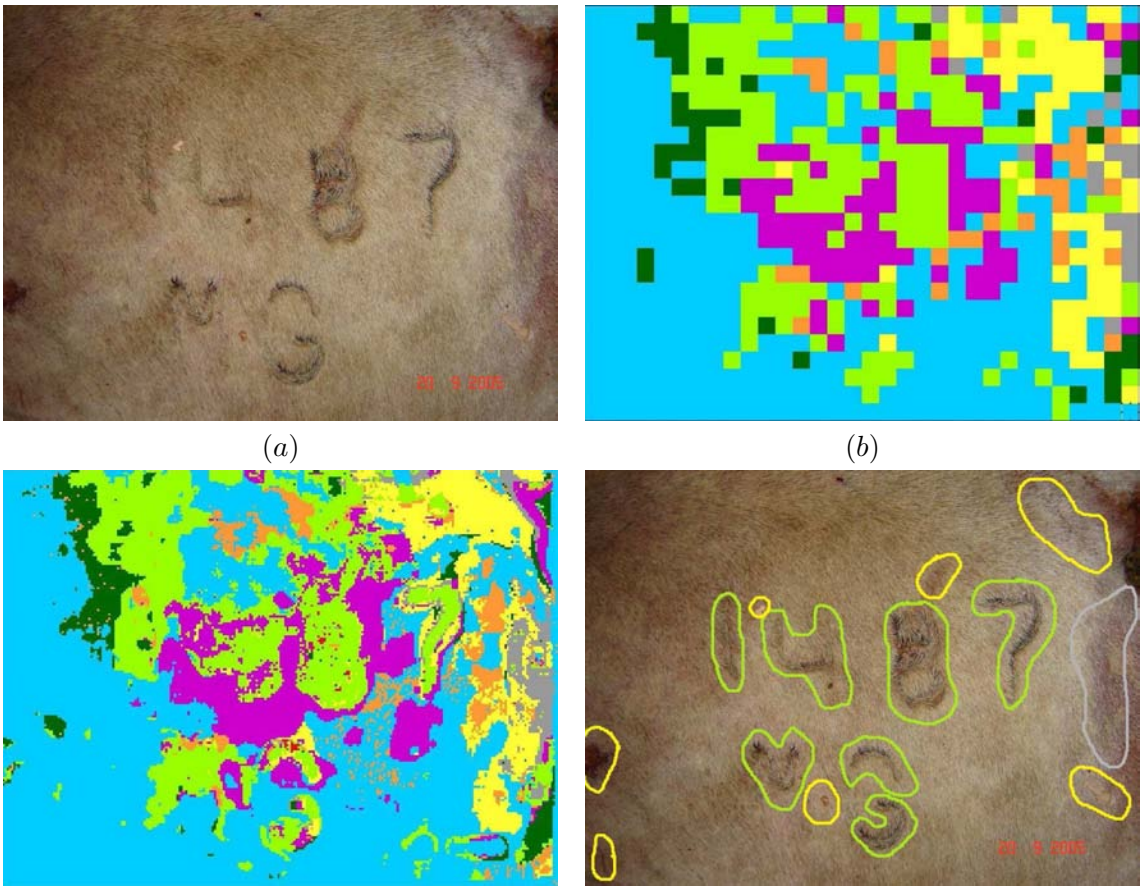
Outro fator interessante que ocorreu foi a classificação de regiões corretas que não foram inseridas na base de aprendizagem. Isso mostra que se aumentarmos e melhorarmos ainda mais a base de imagens, isto é, identificar as características únicas de cada defeito e conseguir representar essas informações ao classificador poderá melhorar ainda mais, a visão do sistema em termos de reconhecimentos de peças de couros não utilizadas para classificação.



(*a*) (*b*)

(*c*) (*d*)

Figura 6.13: Imagem Wet-Blue, utilizada para testes utilizando o módulo de classificação automática do DTCOURO. (a) Imagem Original, (b) Imagem classificada pelo sistema com menor resolução, (c) Imagem classificada pelo sistema com maior resolução, (d) Imagem classificada manualmente.



(*a*) (*b*)

(*c*) (*d*)

Figura 6.14: Imagem Couro-Cru, utilizada para testes utilizando o módulo de classificação automática do DTCOURO. (a) Imagem Original, (b) Imagem classificada pelo sistema com menor resolução, (c) Imagem classificada pelo sistema com maior resolução, (d) Imagem classificada manualmente.

Capítulo 7

Conclusão

A detecção de defeitos e a classificação do couro bovino, com base em inspeção visual, é uma importante etapa da cadeia produtiva do boi. A automatização desse processo, através de sistemas computacionais pode torná-lo mais preciso menos subjetivo, menos sujeito a falhas e mais uniforme. Com base nos primeiros experimentos realizados para análise da classificação do couro bovino conseguimos verificar a complexidade de classificação para esse tipo de textura, onde a maior dificuldade está na classificação de imagens no processo Wet-Blue. Isso ocorreu pelo fato de termos utilizado uma maior quantidade de imagens e que para alguns tipos de defeitos, como: carrapato, berne, risco e furo, existem grande semelhança entre eles, dificultando assim sua discriminação.

Nos experimentos de redução de atributos utilizando a técnica tradicional LDA conseguimos como resultado final, desempenho razoável em termos classificação correta, mostrando que a partir de um conjunto de atributos podemos reduzir para um subconjunto, mantendo a eficiência na classificação. Conseguimos também verificar que à medida que a quantidade de atributos era acrescentada, a taxa de classificação melhorava, até certo nível em que ocorria uma estabilidade.

Com base nos resultados de tempo de treinamento e classificação na redução utilizando FLDA conseguimos concluir que é de suma importância a escolha de um bom classificador, onde casos comprovaram que dependendo da técnica, como $NaiveBayes$ pode haver uma queda substancial de desempenho. Concluímos também que as melhores técnicas para classificação automática do couro bovino foi o IBk, mas concluímos também, que se levarmos em consideração, o custo em relação a tempo de aprendizagem e classificação, a técnica C4.5 é a mais utilizável em termos práticos.

Com os experimentos de simulação de casos de singularidade verificamos que em situações onde temos uma quantidade baixa de amostras e número elevado de atributos, como o reconhecimento de faces poderá ocorrer a instabilidade da matriz de dispersão intra-classes, provocando assim sua singularidade.

Junto com os experimentos comparativos das técnicas propostas que tentam resolver o problema de singularidade identificamos comportamentos similares a técnica tradicional LDA, mas na maioria dos casos tivemos um resultado superior de classificação. Chegamos como resultados desses experimentos que as melhoras técnicas de redução de atributos, em problemas de classificação do couro bovino, tanto em casos que ocorrem ou não a singularidade são as técnicas CLDA e DLDA, e identificamos que as técnicas FisherFace, YLDA e KLDA, tiveram os piores resultados.

Concluímos também que isso pode ser influenciado pela forma de análise que cada técnica utiliza para encontrar as informações mais discriminantes, em que CLDA e DLDA diferente das técnicas FisherFace e YLDA, não removem o espaço nulo da matriz de dispersão intra-classes, a qual a maioria dos trabalhos correlatos aqui pesquisados destacam como o principal problema em sua instabilidade.

Encontramos diversos problemas no uso da técnica KLDA, onde só foi possível reduzir atributos para casos em que existia uma quantidade baixa de amostras, como foi o caso para experimentos de simulação do problema de singularidade. Seus resultados de classificação mostraram a complexidade de classificação após a redução de atributos, chegando ao pior caso tanto para couro-cru quanto wet-blue. Mas sabemos também que a sua implementação não é baseada diretamente no tratamento em situações de instabilidade da matriz de dispersão intra-classes, o que pode ser a maior justificativa para sua baixa taxa de classificação. O grande ganho na realização de experimentos utilizando técnicas baseadas em Kernel foi o conhecimento adquirido sobre seus conceitos e situações para aplicação. Através desses resultados e novas pesquisas realizadas sobre o assunto encontramos diversos trabalhos como o de [18], que apresentam melhorias sobre a técnica de Kernel, utilizando conceitos semelhantes as soluções aqui apresentadas para o problema de singularidade.

Foi também verificado que os módulos implementados facilitaram os experimentos para testes de classificação, reforçando as expectativas de que será possível a implantação de um classificador automático de couros bovinos. Pode ser verificado que a baixa quantidade de imagens para o couro cru favoreceu o alto nível de acerto do classificador, destacando assim que seria necessário um conjunto maior de imagens de couro cru em diferentes ambientes. Para solucionar esse problema, já estão sendo feitas mais visitas junto a Embrapa em diversos frigoríficos, para a captura dessas imagens em posições ideais para outros tipos de experimentos.

Nos testes realizados utilizando o módulo de classificação automática com marcação visual foi possível analisar a dificuldade de classificação de imagens, que não estão presentes no conjunto de treinamento, mas chegando a resultados bem próximos da classificação pelo especialista na área. O grande motivo que levou a classificação de regiões no couro, com defeitos incorretos, está na baixa representatividade dos tipos de defeitos, onde sabemos que para alguns tipos existem diferentes formas visuais para sua classificação.

Através dos resultados obtidos conseguimos concluir o quanto é importante a utilização de técnicas para redução de atributos, em casos que a quantidade de amostras e atributos elevados, influenciando assim diretamente no custo de processamento. Os resultados mostraram a eficiência das técnicas de extração de atributos, utilizados pela biblioteca *Sigus*, para criar as bases iniciais de aprendizagem e a eficiência da ferramenta utilizada para aprendizagem de máquina e realização de experimentos $WEKA$. Um grande ponto importante que ajudou muito no desenvolvimento desse trabalho foi o uso de duas linguagem de programação *Scilab* e *Java*, que deram velocidade e maior estabilidade no desenvolvimento das técnicas aqui apresentadas. Outra grande contribuição desse trabalho está na disponibilização desses códigos, que poderão ajudar futuros trabalhos na área de redução de atributos.

Através desse trabalho conseguimos chegar a diversas conclusões no uso das técnicas de redução de atributos, desde o seu tipo e forma de aplicação. Identificamos também quais as dificuldades de análise do couro mostrando até mesmo a complexidade de classificação de cada fase de tratamento. Mostramos o comportamento de cada técnica de redução de atributos, análise desde a taxa de classificação correta até mesmo o custo de processamento dessas informações. Mostramos o desempenho de redução quando aplicado sobre casos onde ocorrem

singularidade. Identificamos dentre algumas técnicas de aprendizagem, quais foram as melhores e quais são relevantes quando se tratar de desempenho e velocidade de classificação e por fim comparamos a classificação automática de defeitos do couro, com a classificação realizada por um especialista. Com isso podemos observar a complexidade desse problema e quanto é importante a pesquisa e desenvolvimento de constantes melhorias, tanto nas técnicas de redução quanto do sistema de classificação automática, para que no futuro possamos ter um sistema robusto, que possa assim, realizar o seu principal objetivo, que é apoiar as pessoas que trabalham na área a ter mais precisão e velocidade no seu trabalho de classificação de couros bovinos.

Anexo A

Conceitos de Álgebra

Métodos estatísticos multivariados são baseados em manipulação de matrizes. Uma ferramenta básica para o entendimento dessa área é a álgebra. A seguir serão apresentados alguns tópicos dessa área acompanhados de exemplos, que foram utilizados no entendimento da análise discriminante de Fisher. Como base para entendimento para alguns termos relacionados a matriz, iremos descrever brevemente conceitos básicos, com o único objetivo de facilitar o entendimento dos demais termos.

A.1 Matriz

Matriz é considerado no modo algébrico como um arranjo bidimensional formado por elementos (i, j), em que i é a representação da linha e j representação da coluna. Operações com matrizes podem ser utilizadas como uma forma compacta de realizar operações com vários números simultaneamente. Na ilustração A.1.1 é mostrado um exemplo de matriz $M(i, j)$.

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \tag{A.1.1}$$

Uma matriz pode ser quadrada ou não. A quadrada é quando o número de linhas é igual ao número de colunas, e a forma não quadrada se baseia no número de linhas diferente do número de colunas. A matriz A.1.2 ilustra a matriz quadrada e A.1.3 ilustra a matriz não quadrada.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \tag{A.1.2}$$

$$\mathbf{Y} = \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \end{pmatrix} \tag{A.1.3}$$

Uma matriz contendo somente uma linha é conhecida como vetor linha e quando contém somente uma coluna é chamada de vetor coluna. Em A.1.4 e A.1.5, são ilustrados esses dois tipos de situações.

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \tag{A.1.4}$$

$$\mathbf{Y} = \begin{pmatrix} y_{11} \\ y_{21} \\ \vdots \\ y_n \end{pmatrix} \tag{A.1.5}$$

Em A.1.6 é apresentado uma matriz diagonal a partir da matriz quadrada, em que os elementos fora da diagonal principal são todos iguais a zero. Matriz identidade ou matriz unitária é uma matriz quadrada em que os elementos da diagonal principal são todos iguais a 1 e os demais igual a 0. A matriz A.1.7 ilustra a matriz identidade.

$$\mathbf{A} = \begin{pmatrix} x_{11} & 0 & 0 \\ 0 & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix} \tag{A.1.6}$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.1.7}$$

A.1.1 Determinante

O determinante de uma matriz é muito utilizado em resolução de sistemas lineares. Para se calcular a determinante de uma matriz, podem ser utilizados os seguinte métodos ilustrados em A.1.9 e A.1.11.

$$\begin{pmatrix} A \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \tag{A.1.8}$$

$$det(A) = a_{11}a_{22} - a_{21}a_{12} \tag{A.1.9}$$

$$\begin{pmatrix} B \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \tag{A.1.10}$$

$$det(B) = b_{11}b_{22}b_{33} + b_{21}b_{32}b_{13} + b_{31}b_{12}b_{23} - b_{11}b_{32}b_{23} - b_{21}b_{12}b_{33} - b_{31}b_{22}b_{13} \tag{A.1.11}$$

Existem diversas propriedades que auxiliam no cálculo de determinando de uma matriz: (1) Os determinantes de uma matriz e de sua transposta são iguais, isto é, $det(A^T) = det(A)$; (2) Se uma matriz B é obtida de uma matriz A trocando-se duas linhas (ou colunas) de A, então $det(B) = -det(A)$; (3) Se duas linhas (ou colunas) de A são iguais, então $det(A) = 0$; (4) Se A tem uma linha (ou coluna) nula, então $det(A) = 0$; (5) Se B é obtida de A multiplicando-se uma linha (ou coluna) de A por um número real c, então $det(B) = c.det(A)$; (6) O determinante de um produto de matrizes é igual ao produto de seus determinantes, isto é, $det(AB) = det(A)det(B)$.

A.1.2 Matriz Inversa

O inverso de uma matriz (A) é representado por $(A)^{-1}$ e seu resultado deve satisfazer a condição de $(A) * (A)^{-1} = (I)$. Para alguns ocorre problemas como matriz invertível, onde a matriz inversa não existe. Para calcular a inversa de uma matriz, basta estender a matriz (A) com a matriz identidade e aplicar em suas linhas e colunas as operações necessárias para que (A) se torne a matriz identidade. A ilustração A.1.12, mostra um exemplo de cálculo da inversa de uma matriz [10].

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1|1 & 1 & 1 \\ 1 & 1 & 1|1 & 1 & 1 \\ 2 & 3 & 2|1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -2 & 1 \\ -1 & 4 & -1 \end{pmatrix} \tag{A.1.12}$$

Para verificar se uma matriz quadrada é invertível, ou seja, se admite inversa, seu determinante deve ser diferente de *zero*, como é ilustrada em A.1.13.

$$(A)^{-1} = \begin{cases} \exists, & \text{se} \quad det(A) \neq 0 \\ \nexists, & \text{se} \quad det(A) = 0 \end{cases} \tag{A.1.13}$$

A.1.3 Matriz de Variância e Covariância

A matriz de variância e covariância é utilizada para calcular a dispersão estatística entre variáveis. A variância de um vetor $x = [x_1, x_2, ..., x_n]^t$, mede o afastamento dos seus elementos em torno da média (μ_x). A covariância é calculada de forma análoga à variância, mas envolvendo dois vectores de igual dimensão. Da mesma forma, sejam $x_1, ..., x_n$, variáveis aleatórias com variância ${\sigma^2}_1 ... {\sigma^2}_n$ e covariâncias $\sigma_{12}, \sigma_{13}, ..., \sigma_{(k-1)k}$, temos que, ${\sigma^2}_i = E[(x_i - E(x_i))(x_j - E(x_j))]$, para $i \neq j$.

Agrupando as variâncias e covariâncias em uma matriz, temos a seguinte matriz A.1.14, que é chamada de matriz de variância e covariância ou matriz de dispersão das variáveis aleatórias $x_1, \ldots, x_n$. A matriz de variância e covâriancia é simétrica, isto é, $V^t = V$ e o elemento de posição (i, i) é a variância da variável x_i e o elemento de posição (i, j), para $i \neq j$, é a covariância, entre as variáveis x_i e x_j. Assim podemos expressar V como $V = Var(X) = E[(X - E(X))(X - E(X))^t]$.

$$\mathbf{Var}(\mathbf{X}) = V = \begin{pmatrix} {\sigma^2}_1 & \sigma_{12} & \ldots & \sigma_{1n} \\ \sigma_{12} & {\sigma^2}_2 & \ldots & \sigma_{2n} \\ \vdots & \vdots & & \vdots \\ \sigma_{1n} & \sigma_{2n} & \ldots & {\sigma^2}_n \end{pmatrix} \tag{A.1.14}$$

A.1.4 Auto-Valores e Auto-Vetores

Para um vetor coluna não nulo X de uma matriz quadrada A, é considerada um autovetor, se existir um escalar λ tal que,

$$AX = \lambda X \leftrightarrow AX = \lambda IX \leftrightarrow (A - \lambda I)X = 0. \tag{A.1.15}$$

Através da solução da equação característica da matriz A.1.16 de ordem nxm, podemos encontrar os n auto-valores de A, podendo assim ser reais ou complexos, inclusive nulos.

$$det(A - \lambda I) = 0 \leftrightarrow \quad \textbf{det} \begin{pmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} - \lambda \end{pmatrix} \tag{A.1.16}$$

Uma vez obtido um dado auto-valor, este poderá ser substituído em A.1.15 e então a equação poderá ser resolvida para obter-se o respectivo auto-vetor. Portanto, haverá n auto-vetores. A expressão $det(A - \lambda I)$ é conhecida como polinômio característico.

Propriedades dos Auto-valores e Auto-vetores

1. A soma dos auto-valores de uma matriz é igual a soma dos elementos de sua diagonal principal;
2. Auto-vetores correspondentes a diferentes auto-valores são linearmente independentes;
3. Uma matriz é singular e somente se contiver um auto-valor nulo;
4. Se X é um auto-vetor de A correspondente ao auto-valor λ, e se A for inversível, então X é um um auto-vetor de A^{-1}, correspondente ao auto-valor $1/\lambda$;
5. Se X for um auto-vetor de A, então kX também será para qualquer $k \neq 0$, e ambos X e kX correspondem ao mesmo auto-valor λ;
6. Uma matriz e sua transposta possuem os mesmos auto-valores;
7. Os auto-valores de uma matriz triangular superior ou inferior são os valores de sua diagonal principal;
8. O produto de todos os auto-valores de uma matriz, considerando sua multiplicidade, é igual ao determinante desta matriz;
9. Se X for um auto-vetor de A correspondente ao auto-valor λ, então X é um auto-vetor de $(A - cI)$, correspondente ao auto-valor $\lambda - c$, para qualquer escalar c.

A.1.5 Produto Escalar

O cálculo de produto escalar é baseado na operação entre dois vetores de iguais dimensões resultando assim em um valor escalar. Dados dois vetores $a = [a_1, a_2, ..., a_n]$ e $b = [b_i, b_2, ..., b_n]$, seu produto escalar é dado pela equação A.1.17.

$$a.b = \sum_{i=1}^{n} a_1.b_1 + a_2.b_2 + ... + a_n.b_n, \tag{A.1.17}$$

Em que $\sum$ representa a notação de somatória e n o número de dimensões dos vetores referente ao cálculo. Para entendermos melhor, considere o seguinte exemplo. Considere dois vetores de

3 dimensões que são $a = [1, 3, -5]$ e $b = [4, -2, -1]$, o produto escalar é calculado da seguinte forma em A.1.18.

$$[1, 3, -5].[4, -2, -1] = (1)(4) + (3)(-2) + (-5)(-1) = 3. \tag{A.1.18}$$

Definimos então o produto escalar de dois vetores da seguinte forma em A.1.19.

$$a.b = \sum \bar{b_i}.a_i. \tag{A.1.19}$$

Em que $\bar{b_i}$ é complex conjugate de b_i, sendo $a.b = \bar{b.a}$. O produto escalar é tipicamente aplicado para espaço de vetores ortogonais.

Conversão para multiplicação de matrizes

Em casos de multiplicação de matrizes ou vetores com dimensões de $nx1$, o produto escalar pode ser escrito da seguinte maneira, como descrita em A.1.20.

$$a.b = a^T b. \tag{A.1.20}$$

Em que a^T representa a transposta da matriz a, que nesse caso específico temos que a é uma matriz coluna, sendo a transposta de a igual a um vetor linha de (1xn). Como exemplo podemos considerar o seguinte caso. O produto escalar de dois vetores, sendo eles A.1.21 e A.1.22 é equivalente ao produto de uma matriz $1x3$ por $3x1$, em que resultará em um valor escalar de $1x1$, como descrito em A.1.23.

$$\mathbf{a} = \begin{pmatrix} 1 & 3 & -5 \end{pmatrix} \tag{A.1.21}$$

$$\mathbf{b} = \begin{pmatrix} 4 \\ -2 \\ -1 \end{pmatrix} \tag{A.1.22}$$

$$a.b = [3] \tag{A.1.23}$$

A.1.6 Combinação Linear

Uma das características mais importantes de um espaço vetorial é a obtenção de novos valores e novos vetores a partir de um conjunto pré conhecido de vetores desse espaço. Uma expressão da forma $a_1u_1 + a_2u_2 + ... + a_n = w$, em que $a_1, a_2, ..., a_n$ são escalares e $u_1, u_2, ..., u_n$ e w, vetores do $\Re^n$ é chamado de combinação linear.

Em outras palavras, sejam V um espaço vetorial real, $v_1, v_2, ..., v_n \epsilon V$ e $a_1, ..., a_n$, números $\Re$ (ou complexos), então o vetor é um elemento de V, e dizemos que v é uma combinação linear de $v_1, ..., v_n$ resultando no sub-espaço W.

Por exemplo, dados os vetores $v_1 = (1, 0, 0)$, $v_2 = (0, 1, 0)$ e $v_3 = (0, 0, 1)$, esses mesmos, geram um espaço vetorial R^3, por que qualquer vetor (a, b, c) ϵ R^3 pode ser escrito como combinação linear dos v_i, especificamente, $(x, y, z) = x(1, 0, 0) + y(0, 1, 0) + z(0, 0, 1)$.

A.1.7 Dependência e Independência Linear

Podemos dizer que um conjunto de vetores é linearmente dependente, se for possível expressar um dos vetores como uma combinação linear dos outros vetores, como: $v_i = \alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_{i-1} v_{i-1} + \alpha_{i+1} v_{i+1} + ... + \alpha_n v_n$, em que os $\alpha_{i's}$ são escalares. Um conjunto é linearmente independente se ele não é linearmente dependente.

Definição. Seja V um espaço vetorial e $v_1, v_2, ..., v_n \epsilon V$. Dizemos que o conjunto $A = v_1, v_2, ..., v_n$ é linearmente independente (LI) ou os vetores $v_1, v_2, ..., v_n$ são L.I. se a equação: $\alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_n v_n = 0$ e $\alpha_1, \alpha_2, ..., \alpha_n \epsilon \Re$, o que implica em $\alpha_1 = 0, \alpha_2 = 0, ..., \alpha_n = 0$.

Caso exista algum $\alpha_i \neq 0$, dizemos então que $v_1, v_2, ..., v_n$ é linearmente dependente (LD), ou que os vetores $v_1, v_2, ..., v_n$ são linearmente dependentes.

Propriedades da Dependência e Independência Linear

1. Pela definição, o conjunto vazio é linearmente independente.
2. Todo conjunto que contém o vetor nulo é linearmente dependente.
3. Todo conjunto que tem um subconjunto linearmente dependente é linearmente dependente.
4. Todo subconjunto de um conjunto linearmente independente é linearmente independente.
5. Se um vetor de um conjunto é combinação linear de outros vetores desse conjunto, então o conjunto é linearmente dependente.
6. A interseção de dois conjuntos linearmente independentes é linearmente independente - podendo ser o conjunto vazio.
7. A interseção de um número qualquer de conjuntos linearmente independentes é linearmente independente.

A.1.8 Diagonalização de Matriz

A diagonalização de uma matriz é um processo de análise para conversão de uma matriz quadrada em uma matriz de formato especial chamada matriz diagonal [1]. O processo de diagonalização de uma matriz é equivalente a transformar um sistema de equações subjacentes em um conjunto de eixos de coordenadas que transforma uma matriz em forma canônica. Uma diagonalização de matriz é também semelhante a busca por auto-valores de matrizes a qual mais precisamente se torna a saída de uma matriz diagonalizada.

A relação entre a diagonalização de matrizes, auto-valores e auto-vetores, decorre da identidade (decomposição auto-valor) que uma matriz quadrada A pode ser decomposta em uma forma especial, como mostrada na fórmula A.1.24.

[1] Matriz Diagonal é uma matriz quadrada em que os elementos fora da diagonal são zero.

$$A = PDP^{-1}, \tag{A.1.24}$$

onde P é uma matriz composta de auto-vetores de A, D é a matriz diagonal construída sobre os auto-valores correspondentes e P^{-1} é a matriz inversa de P. De acordo com o teorema de decomposição (auto-valor), temos a equação A.1.25.

$$AX = Y, \tag{A.1.25}$$

que também pode ser sempre escrita da forma da equação A.1.26,

$$PDP^{-1}X = Y, \tag{A.1.26}$$

caso ambos os lados forem matrizes quadradas, multiplicando por P^{-1} temos a equação A.1.27.

$$DP^{-1}X = P^{-1}Y, \tag{A.1.27}$$

Uma vez que a transformação linear P^{-1} esteja sendo aplicada em ambos X e Y, significa que resolvendo o sistema original será equivalente em resolver a solução do sistema transformado da equação A.1.28.

$$DX' = Y' \tag{A.1.28}$$

onde $X' \equiv P^{-1}X$ e $Y' \equiv P^{-1}Y$.

Isso conseqüentemente fornece uma maneira mais simples canônica de resolver o sistema, reduzindo o número de parâmetros de nxn para uma matriz arbitraria n da matriz diagonal, obtendo as características da matriz inicial.

Referências Bibliográficas

[1] C. Amado and A. M. Pires. *Uso de Técnicas de Reamostragem para Selecção de Variáveis em Análise Discriminante.* In M. Souto de Miranda and I. Pereira (editors), Estatística: a diversidade na unidade, Novas Tecnologias/Estatística, Edições Salamandra, 1998.

[2] W. P. AMORIM and H. PISTORI. Análise discriminante de fisher aplicadas à detecção de defeitos em couro bovino. *III WVC - Workshop de Visão Computacional, 2007, São José do Rio Preto. Anais do III WVC - Workshop de Visão Computacional*, 2007.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analisys and Machine Intelligence*, 1997.

[4] A. Beluco, A. Buleco, and P. M. Engel. *Classificação de imagens de sensoriamento remoto baseada em textura por redes neurais.* Anais XI SBSR, 2006.

[5] T. E. Campo. Técnicas de seleção de características com aplicações em reconhecimento de faces. Master's thesis, Universidade de São Paulo - USP, Instituto de Matemática e Estatística - Ciência da Computação, 2001.

[6] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new lda-based face recognition system which can solve the small sample size problem. 33(10):1713–1726, October 2000.

[7] J. C. Felipe and A. J. M. Traina. Utilizando características de textura para identificação de tecidos em imagens médicas. *2º Workshop de Informática Médica (WIM), Gramado - RS*, 2002.

[8] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, May 2003.

[9] M. Fukumi and Y. Mitsukura. Feature generation by simple-flda for pattern recognition. *cimca*, 2:730–734, 2005.

[10] B. Gnecco, R. M. Moraes, L. S. Machado, and M. C. Cabral. *Um Sistema de Visualização Imersivo e Interativo de Apoio ao Ensino de Classificação de Imagens.* In: Anais do Symposium on Virtual Reality. Outubro, Florianópolis /SC - Brasil, 2001.

[11] A. Gomes. Aspectos da cadeia produtiva do couro no brasil e em mato grosso do sul. *Embrapa Gado de Corte*, pages 61–72, 2002.

[12] A. Gomes. Avaliação técnica e operacional do sistema de classificação do couro bovino. *Embrapa Gado de Corte, Relatório Final do Projeto (CNPQ)*, 2005.

[13] R. S. Ido Michels, Francisco Bayardo. *Couro Bovino.* Editora UFMS, 2003.

[14] Y. Ji, K. H. Chang, and C.-C. Hung. *Efficient edge detection and object segmentation using Gabor filters*. ACM Press, New York, USA, 2004.

[15] R. Jobanputra and D. Clausi. Texture analysis using gaussian weighted grey level co-occurrence probabilities. In *CRV04*, pages 51–57, 2004.

[16] K. Krastev, L. Georgieva, and N. Angelov. Leather features selection for defects' recognition using fuzzy logic. In *CompSysTech '04: Proceedings of the 5th international conference on Computer systems and technologies*, pages 1–6, New York, NY, USA, 2004. ACM.

[17] W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space approach of fisher discriminant analysis for face recognition. In *In Proc. European Conference on Computer Vision, Biometric Authentication Workshop*, pages 32–44. Springer, 2004.

[18] W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space-based kernel fisher discriminant analysis for face recognition. *Microsoft Research Asia, Beijing Sigma Center, Beijing*, 2004.

[19] E. Martins, P. Azevedo-Marques, L. Oliveira, R. Pereira Jr, and C. M. Trad. Caracterização de lesões intersticinais de pulmão em radiograma de tórax utilizando análise local de textura. *Radio Bras*, 2005.

[20] J. A. Olivera, L. V. Dutra, and C. D. Renn. Aplicações de métodos de extração e seleção de atributos para classificação de regiões. *Instituto Nacional de Pesquisas Espaciais - INPE*, 2005.

[21] M. Pietikainen, T. Maenpaa, and T. Ojala. Gray scale and rotation invariant texture classification with local binary pattern. *In Computer Vision, Sixth European Conference on Computer Vision Proceedings, Lecture Notes in Computer Science 1842*, 420. Springer.:404, 2000.

[22] H. Pistori. Plataforma de apoio e desenvolvimento de sistema guiadas por sinais visuais. *http://www.gpec.ucdb.br/sigus*, 2004.

[23] H. Pistori, W. P. Amorim, M. C. Pereira, P. S. Martins, M. A. Pereira, and M. A. C. Jacinto. Defect detection in raw hide and wet blue leather. *CompIMAGE - Computational Modelling of Objects Represented in Images: Fundamentals, Methods and Applications*, 2006,.

[24] H. Pistori and M. C. Pereira. Integração dos ambientes livres weka e imagej na construção de interfaces guiadas por sinais visuais. In *Anais do V Workshop de Software Livre - WSL*, Porto Alegre, RS, 2-5 de Junho 2004.

[25] L. Ramirez, W. P. Amorim, and P. S. Martins. Comparação de matrizes de co-ocorrência e mapas de interação na detecção de defeitos em couro bovino. *Workshop de Iniciação Científica - SIBGRAPI*, 2006.

[26] S. Russel and P. Norvig. *Inteligência Artificial*. Campus, 2003.

[27] J. C. Santos, J. R. F. Oliveira, and L. V. Dutra. Uso de algoritmos genéticos na seleção de atributos para classificação de regiões. *Anais XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil*, 2007.

[28] O. J. S. Santos and A. Z. Milioni. Composição de especialistas locais para classificação de dados. *Instituto Tecnológico de Aeronáutica (ITA) - Divisão de Engenharia Mecânica-Aeronáutica - São José dos Campos - Brasil*, 2005.

[29] W. R. Schwartz and H. Pedrini. Métodos para classificação de imagens baseada em matrizes de co-ocorrência utilizando características de textura. *Anais do III Colóquio Brasileiro de Ciências Geodésicas*, 2002.

[30] M. P. S. Silva. Mineração de dados : Conceitos, aplicações e experimentos com weka. *Livro da Escola Regional de Informática Rio de Janeiro - Espírito Santo*, 2004.

[31] J. L. Sobral. Leather inspection based on wavelets. In *IbPRIA (2)*, pages 682–688, 2005.

[32] J. A. Souza. *Reconhecimento de Padrões usando Indexação Recursiva.* PhD thesis, Universidade Federal de Santa Catarina, Departamento de Engenharia de Produção e Sistemas, 1999.

[33] S. Theodoridis and K. Koutroumbas. *Pattern recognition.* Academic press, 1999.

[34] C. Thomaz, E. Kitani, and D. Gillies. A Maximum Uncertainty LDA-based approach for Limited Sample Size problems - with application to Face Recognition. *Journal of the Brazilian Computer Society*, 12(1), June 2006.

[35] J. Yang and J. Yang. *Why can LDA be performed in PCA transformed space?* Pattern Recognition, 2003.

[36] J. Yang, Y. Yu, and W. Kunz. An efficient lda algorithm for face recognition. In *6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000)*, 2000.

[37] M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGRí02)*, 2002.

[38] J. Ye and T. Xiong. Null space versus orthogonal linear discriminant analisys. *Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania*, 2006.

[39] H. Yu and H. Yang. A direct lda algorithm for high-dimensional data - with application to face recogniion, 2067.

[40] J. Zhang and K. K. Ma. Kernel fisher discriminant for texture classification. *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*, 2004.

[41] W. Zhao, R. Chellappa, and P. J. Phillips. Subspace linear discriminant analisys to face recognition. *Partially supported by the Office of Naval Research*, 1999.