

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Carlos Estevão Bastos Sousa

**Aplicação de Técnicas de Visão Computacional e Aprendizado de
Máquina como Ferramenta de Apoio à Detecção de Falhas e
Definição do Nível de Qualidade em Peças de Couro Caprino**

Fortaleza – CE

2 de dezembro de 2020

Carlos Estevão Bastos Sousa

**Aplicação de Técnicas de Visão Computacional e
Aprendizado de Máquina como Ferramenta de Apoio à
Detecção de Falhas e Definição do Nível de Qualidade
em Peças de Couro Caprino**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Inteligência Artificial

Orientador: Prof. Dr. Cláudio Marques de Sá Medeiros

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

2 de dezembro de 2020



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ

CARLOS ESTEVÃO BASTOS SOUSA

Aplicação de Técnicas de Visão Computacional como Ferramenta de Apoio à
Qualificação de Couro Caprino

Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação, área de concentração Ciência da Computação.

Aprovada em 17/09/2020.

BANCA EXAMINADORA

Prof. Dr. Cláudio Marques de Sá Medeiros (Orientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Pedro Pedrosa Reboucas Filho

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE

Prof. Dr. Antonio Carlos da Silva Barros

Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB

Prof. Dr. Paulo César Cortez

Universidade Federal do Ceará - UFC



Documento assinado eletronicamente por **Pedro Pedrosa Reboucas Filho, Professor do Ensino Básico, Técnico e Tecnológico**, em 23/09/2020, às 10:51, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Antonio Carlos da Silva Barros, Usuário Externo**, em 25/09/2020, às 13:54, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Claudio Marques de Sa Medeiros, Professor do Ensino Básico, Técnico e Tecnológico**, em 28/09/2020, às 15:08, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Paulo César Cortez, Usuário Externo**, em 28/09/2020, às 17:18, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade do documento pode ser conferida no site https://sei.ifce.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 informando o código verificador **1995997** e o código CRC **9B384F9D**.

Este trabalho é dedicado àqueles que, em meio às dificuldades, fazem sempre o seu melhor.

Agradecimentos

Elaborar um texto em agradecimento a todos que contribuíram para a conclusão deste trabalho, provavelmente tornaria este documento muito grande. Diversas são as pessoas que ajudaram direta ou indiretamente durante este período acadêmico. Desta forma, seguem os agradecimentos especiais.

Aos meus pais Maria das Graças e Estevão, pois sempre me apoiaram e contribuíram para o meu crescimento intelectual.

À minha irmã Ana Verônica que sempre foi um grande exemplo para mim.

Ao professor Cláudio Sá, que me orientou incessantemente, mesmo nos momentos de dificuldades. Sou extremamente grato, não só pelos conselhos, paciência e contribuições para o meu crescimento intelectual durante o mestrado, mas também pelos ensinamentos que ecoarão pelo resto da minha vida.

A todos que fazem parte da empresa CV Couros e Peles LTDA, com ênfase às dirigentes Roseane Medeiros e Viviane Oliveira e aos especialistas Ananias Vidal, Valdir e Eudes.

Aos meus colegas de trabalho Núbia, Taylana e Jardel, que me apoiaram durante esta jornada, compreendendo-me nos momentos de ausência e, em especial às professoras Clair e Claudiane pelas ajudas em diversos momentos da construção deste documento.

Ao meu primo Glauco, que me apoiou durante todos os momentos de minha vida, sempre solícito e mostrando o real valor da palavra família.

Ao meu amigo Ismael que disponibilizou local para dormir após as aulas do mestrado realizadas no turno da noite.

A todos os professores do Programa de Pós-graduação em Ciências da Computação (PPGCC) do Instituto Federal de Educação Ciência e Tecnologia do Ceará (IFCE) por todas as experiências vivenciadas e pelos conhecimentos compartilhados, em especial aos professores Ajalmar Rocha e Pedro Pedrosa que sempre foram bastante gentis e prontos a me atender nos momentos de dúvidas.

À minha amada esposa, que demonstra muita paciência durante os meus períodos de estudo, fazendo sempre o seu melhor para contribuir para o futuro de nossa família.

No mais, dedico minha gratidão a todos os meus amigos e colegas que proporcionaram de alguma forma algum aprendizado.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.”
(Marthin Luther King)

Resumo

O Nordeste brasileiro, em relação a criação de caprinos, destaca-se das demais regiões por possuir cerca de 92,82 % de todos os rebanhos existentes no país. Tal valor remete a região citada como um dos principais polos produtores de couro produzido a partir da pele desses animais. No entanto, apesar do exposto, os processos desempenhados para a produção do couro, detecção de defeitos e definição do nível de qualidade ocorrem de maneira totalmente manual e subjetiva, o que os torna lentos, trabalhosos e passíveis de ocorrências de diversos erros. Em pesquisa, pode-se observar que a definição do nível de qualidade do mesmo conjunto de peças de couro por profissionais distintos e de uma mesma empresa, mostra-se com um nível de concordância de aproximadamente 42 %, o que pode acarretar em diversos prejuízos financeiros para este setor. Tendo em vista tal problemática, é proposto um sistema de apoio a decisão que detecte falhas e atribua um nível de qualidade às peças de couro caprino *wet blue*. Para a elaboração deste sistema, são adquiridos conjuntos de imagens, aplicadas técnicas de visão computacional com o intuito de detectar bordas, remover plano de fundo, adquirir objetos de interesse e extrair características e, a partir dessas, formar conjuntos de dados e aplicá-los a técnicas de inteligência artificial baseadas redes neurais, instâncias, probabilidade e vetores de suporte, com o intuito de detectar defeitos e, posteriormente, atribuir um nível de qualidade às peças de couro caprino. Como resultado desta dissertação apenas uma metodologia é escolhida para a detecção de falhas e outra para a definição do nível de qualidade. Deste modo, como melhores resultados obtidos destacam-se a aplicação de objetos de interesse de 51 x 51 *pixels*, extrator *Gray Level Co-Occurrence Matrix* (GLCM) e classificador *Support Vector Machine_{RBF}* que possibilitou o alcance de uma acurácia de 96,47 % no treinamento e 96,31 % no teste para a detecção de falhas. Quanto a qualificação das peças de couro, a mesma ocorre através da localização das maiores regiões quadradas e livres de falhas, a partir dessas são adquiridos atributos e formados conjuntos de dados. Em relação a classificação, é utilizado o classificador *Naïve Bayes*, com o qual pôde-se alcançar uma acurácia de 95,24 % no treinamento e 93,26 % no teste. Por fim, destaca-se que o sistema proposto torna-se adequado para este tipo de indústria, tendo em vista que pode ser utilizado como um sistema de apoio a decisão para a detecção de falhas e qualificação das peças de couro caprino.

Palavras-chave: Detecção de Falhas em Couro, Qualificação de Couro, Sistema de Apoio à Decisão.

Abstract

The Brazilian Northeast, in relation to the creation of goats, stands out from the other regions for having about 92.82% of all the existing herds in the country. This value refers to the region cited as one of the main hubs producing leather produced from the skin of these animals. However, despite the foregoing, the processes performed for the production of leather, detection of defects and definition of the level of quality occur in a totally manual manner and classification, which makes them slow, laborious, and susceptible to the occurrence of several errors. In research, it can be observed that the definition of the quality level of the same set of leather pieces by different professionals and from the same company, shows a level of agreement of approximately 42 %, which can result in several losses financial resources for this sector. In view of this problem, a decision support system that detects flaws and assigns a level of quality to wet blue goat leather is proposed. For the elaboration of the system, sets of images are acquired, computer vision techniques are applied in order to detect edges, remove backgrounds, acquire objects of interest and extract characteristics and, from these, form data sets and apply them the artificial intelligence technique based on neural networks, competence, probability, and support vectors, with the aim of detecting defects and, subsequently, assigning a level of quality to goat leather pieces. As a result of this dissertation, only one methodology is chosen for the detection of failures and another for the definition of the quality level. In this way, as the best-chosen results, the application of objects of interest of 51 x 51 pixels, Gray Level Co-Occurrence Matrix (GLCM) extractor and Support Vector Machine_{RBF} classifier, which made it possible to achieve accuracy 96.47 % in training and 96.31 % in the test for fault detection. As for the qualification of leather parts, it occurs through the location of the largest regions and free of flaws, from these attributes are acquired and data sets are formed. Regarding the classification, the Naïve Bayes classifier is used, with which it was possible to achieve an accuracy of 95.24 % in training and 93.26% in the test. Finally, it is noteworthy that the proposed system becomes suitable for this type of industry, considering that it can be used as a decision support system for the detection of failures and qualification of goat leather pieces.

Keywords: Fault Detection in Leather, Qualification of Leather, Decision Support System.

Lista de ilustrações

Figura 1 – Quantidade de rebanhos por região, valores referentes a 2006 e 2017.	32
Figura 2 – Separação das peles por tamanho.	33
Figura 3 – Fulões.	34
Figura 4 – Representação pictográfica de regiões da pele de caprinos.	35
Figura 5 – Procedimentos metodológicos realizados para a elaboração do sistema	39
Figura 6 – Imagens capturadas durante o treinamento na empresa CV Couros e Peles Ltda.	40
Figura 7 – Peça de couro no lado carnal. Em destaque a falha corte leve de faca.	42
Figura 8 – Exemplo de falha relacionada à cicatriz.	43
Figura 9 – Etiquetas inseridas nas peles.	44
Figura 10 – Parte das amostras prontas para requalificação.	44
Figura 11 – Especialista requalificando as peças de couro.	45
Figura 12 – Mesa para fotografia das amostras.	46
Figura 13 – Captura das amostras de couro caprino disponibilizadas.	46
Figura 14 – Exemplo de localização de falha.	48
Figura 15 – Localização de regiões de falhas.	48
Figura 16 – Exemplo de região de interesse de tamanho 51 x 51 <i>pixels</i>	49
Figura 17 – Exemplo de aplicação do método de validação cruzada <i>K-Fold</i>	53
Figura 18 – Fluxo do pré-processamento da imagem.	55
Figura 19 – Técnica de pré-processamento baseada em Erosão e Dilatação.	56
Figura 20 – Aplicação de dilatação e erosão em uma imagem binarizada.	57
Figura 21 – Técnica de pré-processamento baseada na Detecção de Bordas.	57
Figura 22 – Detecção do contorno e colorização da peça de couro caprino.	58
Figura 23 – Exemplo de divisão em regiões de interesse.	59
Figura 24 – Exemplo da falha buraco em diversos tamanhos de regiões de interesse.	59
Figura 25 – Exemplo da aplicação do algoritmo detector de falhas em regiões de interesse de 101 <i>pixels</i>	60
Figura 26 – Exemplo de imagem processada a partir do uso de 4 núcleos: P1, P2, P3 e P4.	61
Figura 27 – Exemplo de pele desejável (boa) e com diferentes níveis de defeitos.	62
Figura 28 – Binarização e elaboração da matriz de referência.	62
Figura 29 – Processo de detecção das regiões livres de falha.	63
Figura 30 – Técnicas aplicadas para remoção do plano de fundo.	65
Figura 31 – Resultado médio da aplicação do paralelismo.	75
Figura 32 – Destaque de falhas a partir de uma análise a olho nu.	76
Figura 33 – Exemplo de detecção de falhas com regiões de interesse de 31 x 31 <i>pixels</i>	77

Figura 34 – Exemplo de detecção de falhas com regiões de interesse de 41 x 41 <i>pixels</i> .	77
Figura 35 – Exemplo de detecção de falhas com regiões de interesse de 51 x 51 <i>pixels</i> .	77
Figura 36 – Exemplo de detecção de falhas com regiões de interesse de 75 x 75 <i>pixels</i> .	78
Figura 37 – Exemplo de detecção de falhas com regiões de interesse de 101 x 101 <i>pixels</i> .	78
Figura 38 – Resultados médios obtidos da classificação, com base na detecção de 80% das regiões livres de falhas.	79
Figura 39 – Limiarização com diversos parâmetros distintos para T .	100
Figura 40 – Fragmento de imagem.	101
Figura 41 – Filtro da média aplicado em um fragmento de imagem.	101
Figura 42 – Filtro da média aplicado em um fragmento de imagem relacionado à borda de uma pele.	102
Figura 43 – Fragmento de imagem.	102
Figura 44 – Filtro da média aplicado em um fragmento de imagem.	103
Figura 45 – Filtro da mediana aplicado em um fragmento de imagem relacionado à borda de uma pele.	103
Figura 46 – Aplicação dos operadores de Sobel.	104
Figura 47 – Processo de crescimento de região.	105
Figura 48 – Processo de crescimento de região em uma imagem binária.	106
Figura 49 – Exemplo da aplicação da erosão.	107
Figura 50 – Aplicação do método de erosão em uma imagem limiarizada de amostra de couro caprino.	107
Figura 51 – Exemplo da aplicação da dilatação.	108
Figura 52 – Aplicação de dilatação e seus elementos estruturantes em um fragmento de imagem relacionado à borda de uma peça de couro.	108
Figura 53 – Exemplo de cálculo da matriz GLCM.	109
Figura 54 – Ângulos utilizados para o cálculo da matriz de co-ocorrência	110
Figura 55 – (a) Fragmento de imagem, (b) construção de uma matriz de co-ocorrência ($C_{d\theta}$), (c) $C_{d\theta}$ simétrica ($d = 1, \theta = 90^\circ$) (d) $C_{d\theta}$ assimétrica ($d = 1, \theta = 0^\circ$)	110
Figura 56 – Funcionamento do método LBP.	112
Figura 57 – Exemplos de vizinhanças circulares, (a) $LBP_{1,8}$; (b) $LBP_{2,16}$ e $LBP_{2,8}$.	113
Figura 58 – Aplicação do método LBP com diferentes tamanhos de raio e número de pontos.	113
Figura 59 – Estrutura do neurônio biológico.	115
Figura 60 – Estrutura de um neurônio artificial.	116
Figura 61 – Representação do algoritmo <i>Logistic Regression</i> .	117
Figura 62 – Função sigmóide logística.	118
Figura 63 – Representação do funcionamento do algoritmo KNN para $K = 3$.	119

Figura 64 – Representação do funcionamento do algoritmo KNN.	119
Figura 65 – Representação de uma Rede Neural Multilayer Perceptron.	122
Figura 66 – Estrutura de uma RNA do tipo MLP com o algoritmo <i>backpropagation</i>	123
Figura 67 – Exemplo de hiperplano ótimo em um conjunto de dados linearmente separável.	124
Figura 68 – Exemplo de aplicação do parâmetro C.	125
Figura 69 – Exemplo de conjunto de dados linear (a) e não linear(b).	125
Figura 70 – Aplicação do truque de kernel em um conjunto de dados não linear.	126
Figura 71 – Problema sendo solucionado de forma sequencial.	127
Figura 72 – Processador <i>Quad-core Intel Sandy Bridge</i>	128
Figura 73 – Problema sendo solucionado de forma paralela.	128
Figura 74 – Exemplo de uma imagem digital contendo defeitos (0) e regiões boas (1).	131
Figura 75 – Primeira linha e coluna de uma determinada imagem sendo copiada para uma matriz auxiliar.	131
Figura 76 – Representação pictográfica de regiões da pele de caprinos.	132
Figura 77 – Representação pictográfica de regiões da pele de caprinos.	132

Lista de tabelas

Tabela 1 – Principais falhas encontradas em peças de couro caprino.	40
Tabela 2 – Principais falhas encontradas em peças de couro caprino - continuação.	41
Tabela 3 – Principais falhas encontradas em peças de couro caprino - continuação.	42
Tabela 4 – Quantidade de peles recebidas.	43
Tabela 5 – Matriz de Confusão com o rótulo dos especialistas.	45
Tabela 6 – Agrupamentos elaborados (falhas e regiões normais).	47
Tabela 7 – Exemplo de alguns dados contidos no relatório elaborado.	48
Tabela 8 – Parâmetros utilizados no extrator LBP.	50
Tabela 9 – Conjuntos de dados adquiridos a partir de regiões de falhas.	50
Tabela 10 – Matriz de Confusão	53
Tabela 11 – Melhores resultados médios obtidos a partir do perceptron simples.	67
Tabela 12 – Melhores resultados médios obtidos a partir do <i>Logistic Regression</i>	67
Tabela 13 – Melhores resultados médios obtidos a partir do KNN.	68
Tabela 14 – Melhores resultados médios obtidos a partir do classificador de Fisher.	68
Tabela 15 – Melhores resultados médios obtidos a partir do <i>Naïve Bayes</i>	69
Tabela 16 – Melhores resultados médios obtidos a partir do MLP.	70
Tabela 17 – Melhores resultados médios obtidos a partir do SVM.	70
Tabela 18 – Matriz de confusão proveniente do processo de treinamento e teste com <i>Logistic Regression</i> , dados obtidos a partir do extrator GLCM e regiões de interesse de 21 x 21 <i>pixels</i>	71
Tabela 19 – Matriz de confusão proveniente do processo de treinamento e teste com MLP, dados obtidos a partir do extrator GLCM e regiões de interesse de 31 x 31 <i>pixels</i>	71
Tabela 20 – Matriz de confusão proveniente do processo de treinamento e teste com SVM, dados obtidos a partir do extrator GLCM e regiões de interesse de 21 x 21 <i>pixels</i>	71
Tabela 21 – Valores referentes à precisão, <i>recall</i> e F1 relacionados aos melhores modelos obtidos.	72
Tabela 22 – Análise da distribuição dos erros de classificação durante o treinamento e teste.	72
Tabela 23 – Tempo médio para detecção de falhas em couro - <i>Logistic Regression</i>	74
Tabela 24 – Tempo médio para detecção de falhas em couro - MLP.	74
Tabela 25 – Tempo médio para detecção de falhas em couro - SVM.	75
Tabela 26 – Melhores resultados médios adquiridos durante o processo de qualificação com base na detecção de 80% dos <i>pixels</i> isentos de falhas.	79

Tabela 27 – Melhores resultados médios obtidos durante a redução do número de regiões de interesse a serem analisadas.	80
Tabela 28 – Melhor resultado na qualificação de peças de couro caprino utilizando o classificador <i>Naïve</i> Bayes.	81
Tabela 29 – Análise da acurácia, precisão, <i>recall</i> e F1 do melhor resultado obtido - classificador <i>Naïve</i> Bayes.	81
Tabela 30 – Qualificações definidas pelos classificadores.	94
Tabela 31 – Qualificações definidas pelos classificadores - continuação.	95

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
GLCM	<i>Gray Level Co-Occurrence Matrix</i>
KNN	<i>K-Nearest Neighbors</i>
LBP	<i>Local Binary Patterns</i>
LR	<i>Logistic Regression</i>
MLP	<i>MultiLayer Perceptron</i>
NB	<i>Naïve Bayes</i>
RBF	<i>Radial Basis Function</i>
RNA	Rede Neural Artificial
SLP	<i>Single Layer Perceptron</i>
SVM	<i>Support Vector Machine</i>

Sumário

1	INTRODUÇÃO	27
1.1	Motivação para a Dissertação	28
1.2	Objetivos	29
1.2.1	Objetivos Específicos	29
1.3	Produção Científica	29
1.4	Estrutura da Dissertação	30
2	REVISÃO BIBLIOGRÁFICA	31
2.1	A produção de caprinos no Brasil	31
2.1.1	Processo de Beneficiamento Primário do Couro Caprino	33
2.2	Qualificação do Couro	34
2.3	Trabalhos Relacionados	36
3	METODOLOGIA	39
3.1	Treinamento na Empresa CV Couros e Peles Ltda	39
3.2	Aquisição do Conjunto de Amostras de Couro Caprino	43
3.2.1	Análise da Classificação Visual dos Especialistas	45
3.3	Aquisição do Conjunto de Imagens de Falha	47
3.4	Aquisição dos Conjuntos de Dados para Detecção de Falhas	49
3.5	Organização dos conjuntos de dados e utilização dos algoritmos de aprendizado de máquina	50
3.6	Métricas de avaliação dos classificadores propostos	53
3.7	Detecção de Falhas em Couro Caprino	55
3.7.1	Pré-processamento que antecede a detecção de falhas	55
3.7.1.1	Pré-processamento baseado em crescimento de região	55
3.7.1.2	Pré-processamento baseado em Morfologia Matemática	56
3.7.1.3	Pré-processamento baseado em detecção de bordas	57
3.7.2	Divisão em regiões de interesse	58
3.7.3	Classificação das regiões de interesse	59
3.8	Localização das áreas livres de falhas	61
4	RESULTADOS E DISCUSSÕES	65
4.1	Resultados do pré-processamento das amostras de couro caprino	65
4.2	Resultados obtidos na classificação binária	66
4.2.1	Resultados médios obtidos a partir do treinamento entre amostras normais e amostras de falha	66

4.2.2	Análise dos melhores resultados obtidos pelos classificadores	71
4.3	Análise dos erros de classificação por tipo de falha	72
4.4	Tempo de execução da segmentação de acordo com classificador e	
	dimensão da região de interesse	73
4.4.1	Análise da aceleração média e eficiência na aplicação do paralelismo	75
4.5	Análise da detecção e classificação de regiões de interesse em amos-	
	tras de couro caprino	76
4.6	Resultados obtidos durante o processo de qualificação das peças de	
	couro caprino	78
4.6.1	Resultados obtidos a partir da detecção de 80 % dos <i>pixels</i> sem defeitos . .	79
4.6.2	Resultados obtidos a partir da redução do número de regiões de interesse a	
	serem analisados	80
5	CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS .	83
5.1	Trabalhos Futuros	84

REFERÊNCIAS	85
------------------------------	-----------

APÊNDICES	91
------------------	-----------

APÊNDICE A – QUALIFICAÇÕES DEFINIDAS PELOS ESPECI-	
ALISTAS PARA ELABORAÇÃO DO CONJUNTO	
DE IMAGENS	94

ANEXOS	97
---------------	-----------

	ANEXO A – VISÃO COMPUTACIONAL	99
A.1	Limiarização (<i>thresholding</i>)	99
A.2	Filtros Passa-baixa	100
A.2.1	Filtro da Média	100
A.2.2	Filtro da Mediana	102
A.3	Filtros Passa-alta	103
A.3.1	Operadores de Sobel	104
A.4	Crescimento de Regiões	105
A.5	Operações Morfológicas	106
A.5.1	Erosão	106
A.5.2	Dilatação	107
A.6	Extratores de Características	109

A.6.1	<i>Gray-Level Co-Occurrence Matrix (GLCM)</i>	109
A.6.2	<i>Local Binary Patterns (LBP)</i>	112
	ANEXO B – APRENDIZAGEM DE MÁQUINA	115
B.1	Perceptron de única camada	115
B.2	<i>Logistic Regression</i>	117
B.3	<i>K-Nearest Neighbor (KNN)</i>	118
B.4	Classificador de Fisher baseado na regra de Bayes	120
B.5	Classificador ingênuo de Bayes (<i>Naïve Bayes</i>)	121
B.6	<i>Multilayer Perceptron (MLP)</i>	122
B.7	<i>Support Vector Machine (SVM)</i>	123
	ANEXO C – COMPUTAÇÃO PARALELA	127
	ANEXO D – DETECÇÃO DE ÁREAS LIVRES DE FALHAS	131

1 Introdução

Uma importante alternativa econômica para o semiárido nordestino é representada pela caprino-ovinocultura. Na qual é possível obter, entre outros, a carne, o leite e a pele desses animais, sendo a última bastante apreciada na confecção de artefatos e calçados finos (SANTOS FILHO; BARRETO; DANIEL SANTOS, 2013).

Conforme Cobrasil (2017), na região citada, em função do seu clima, existe um tipo de carneiro que proporciona uma pele singular. Este, chamado no linguajar do ramo de ‘mestiço’, pode ser definido como uma prole proveniente do cruzamento de animais de raças distintas. Possui uma resistência bem superior aos cordeiros de outras regiões e características que permitem a elaboração de um sapato de bico mais fino, resistente e gerando um visual mais elegante.

No entanto, apesar de ser uma alternativa bastante viável para a região, por adaptar-se ao clima semiárido, apresentando uma resistência muito superior aos outros cordeiros, a criação destes animais apresenta-se de maneira bastante rústica. Deste modo, o problema do setor de couros e derivados no Brasil começa na qualidade da matéria prima, a pele (SARETTA, 2012). Isto posto, na maioria das vezes, por não haver um processo de criação de maneira intensiva e confinada, as peles desses animais apresentam bastante defeitos e são facilmente danificadas por ectoparasitas, arranhões causados por espinhos ou cercas de arame farpado.

Além dos fatores citados, o manuseio incorreto durante o abate, esfolamento e conservação das peles prejudicam a qualidade delas, gerando uma depreciação, dada a heterogeneidade dos níveis das peças de couro.

Em contrapartida, produtos de couro industrializados são elaborados a partir de matéria prima de alta qualidade. Deste modo, os curtumes necessitam adquirir e processar peças de couro caprino com o nível esperado pelas indústrias.

Neste sentido, visando adquirir e processar peças de qualidade, os curtumes realizam dois processos de qualificação das peles caprinas. O primeiro ocorre durante a negociação entre a indústria de processamento e os produtores rurais. Neste, as amostras são simplesmente classificadas como boas ou ruins, definindo a partir dela, a aquisição ou não da pele.

No segundo processo de qualificação, após a pele ser adquirida pelos curtumes e passar por várias etapas de processamento até torna-se uma peça de couro denominada *wet blue*, é classificada dentre 7 categorias, na qual a amostra da categoria 1 é considerada a melhor e, à medida que este número aumenta, reduz-se a qualidade dela. Salienta-se que

após esta etapa, as peças de couro são vendidas para as indústrias de acabamento.

Embora a qualificação correta das peças de couro *wet blue* seja de extrema importância, esta operação é atualmente efetuada por inspetores humanos que tendem a passar despercebido um número considerável de defeitos devido o alto grau de subjetividade nesse processo.

Kwak, Ventura e Tofang-Sazi (2001) apresentam que a inspeção visual e classificação dos defeitos de superfície são muito importantes na fabricação de produtos de couro que exigem elevada qualidade. Pelo fato dessas inspeções ocorrerem por humanos, há um grau considerável de erros por parte desses classificadores. Além disso, a inspeção manual e qualificação são processos lentos e trabalhosos. Deste modo, podem se tornar um gargalo crítico em todo o processo produtivo. Assim, é comum a ocorrência de divergências entre os classificadores do curtume sobre a categorização de diversas amostras de couro.

Quando as divergências citadas ocorrem entre o profissional do curtume e o classificador da indústria de acabamento, o impacto financeiro assume proporções bastante indesejáveis.

Tendo em vista a problemática apresentada, nesta Dissertação é proposta a elaboração de um sistema de visão computacional que detecte defeitos em couro caprino *wet blue* e, a partir dessas detecções, atribua um nível de qualidade às peças de couro, produzindo um sistema de apoio à decisão, que visa contribuir para uma qualificação correta e eficaz dentro dos curtumes.

1.1 Motivação para a Dissertação

A primeira e principal motivação está relacionada à colaboração aos curtumes, pois no Ceará não há uma solução adequada e robusta no sentido de buscar minimizar problemas provenientes de erros durante a detecção de falhas e definição do nível de qualidade em peças de couro caprino. Desta forma, visando contribuir no sentido de prover um sistema de apoio à decisão com o intuito de reduzir divergências entre os setores da indústria têxtil.

Na contribuição a este tipo de pesquisa, a partir da elaboração de conjuntos de imagens, pois, conforme Aslam et al. (2019), há uma grande dificuldade em obter conjuntos de dados de couro *wet blue*. Os pesquisadores que trabalham sobre defeitos em couro, geralmente, têm pequenos conjuntos de dados de 60 a 80 imagens. Esses não são suficientes para treinar um modelo de aprendizado.

Por fim, em relação ao estado da arte, com o intuito colaborar com este tipo de pesquisa, a partir da apresentação dos processos que foram introduzidos nesta Dissertação através de investigações relacionadas a trabalhos anteriores, buscando propor novas

metodologias, como também aplicando as que apresentaram melhores resultados.

1.2 Objetivos

O objetivo principal desta Dissertação é propor um sistema de apoio à decisão a partir da avaliação de técnicas inteligência artificial, visão computacional e processamento paralelo aplicadas na aquisição de amostras, da detecção de bordas, formação de conjuntos de imagens, aquisição de conjunto de dados, detecção de falhas e, por fim, atribuição do nível de qualidade de couro peças de couro *wet blue*, seguindo os padrões de qualidade utilizados pelos curtumes.

1.2.1 Objetivos Específicos

Os seguintes objetivos específicos são:

- compreender o processo de fabricação do couro, bem como os tipos de falhas encontrados, a detecção visual delas e a definição do nível de qualidade das peças de couro caprino;
- obter um conjunto de imagens de amostras de couro caprino e a partir delas adquirir um conjunto de imagens de falhas, visando a detecção de defeitos em peças de couro caprino *wet blue*;
- adquirir conjuntos de dados a partir de extratores de características e dos conjuntos de imagens adquiridos;
- implementar e avaliar algoritmos de aprendizado de máquina visando obter o melhor modelo computacional para detecção de regiões de falha em amostras de couro caprino;
- aplicar e avaliar a técnica de multiprocessamento na detecção de falhas em couro caprino, visando reduzir o tempo de execução para este procedimento;
- adquirir atributos a partir de amostras de couro caprino com falhas já detectadas, implementando e avaliando algoritmos de aprendizado de máquina com o intuito de qualificar peças de couro seguindo os padrões de qualidade utilizados pelos curtumes;
- avaliar algoritmos de aprendizado de máquina com o intuito de identificar o nível de qualidade de peças de couro caprino *wet blue*.

1.3 Produção Científica

Durante o projeto de mestrado, os seguintes trabalhos científicos foram aceitos e publicados, a saber:

- Sousa, C. E. B.; Medeiros, C. M. S.; Pereira, R. F.; Neto, M. A. V.; Neto, A. A. Fault detection and qualification of wet blue goatskin. In: *8th Joint Symposium on Computational Intelligence (JSCI8)*. Bangkok, Thailand: IEEE Computacional Intelligence Society Thailand Chapter, 2020. p. 13-17. Disponível em: <<https://site.ieee.org/thailand-cis/files/2019/12/JSCI8-Paper-4.pdf>>
- Sousa, C. E. B.; Medeiros, C. M. S.; Pereira, R. F.; Neto, M. A. V.; Neto, A. A. Defect detection and quality level assignment in wet blue goatskin. In: *Proceedings of the 11th International Conference on Advances in Information Technology*. New York, NY, USA: Association for Computing Machinery, 2020. (IAIT2020). ISBN 9781450377591. Disponível em: <<https://doi.org/10.1145/3406601.3406650>>.
- Sousa, C. E. B.; Medeiros, C. M. S.; Pereira, R. F.; Neto, A. A. Detecção de falhas em couro caprino wet blue a partir da utilização de técnicas de Visão Computacional e Inteligência Artificial. In: *XIV BreSci – Brazilian e-Science Workshop 2020*. Cuiabá-MT: XL Congresso da Sociedade Brasileira de Computação, 2020. Disponível em: <<https://sol.sbc.org.br/index.php/bresci/article/view/11181/11052>>
- Sousa, C. E. B.; Nascimento, L. B. G.; Medeiros, C. M. S.; Oliveira, M.; Trajano, I. A. DECIME: Decision Support Software for Melanoma Skin Cancer Detection. In: *SP4HC - Singular Problems for Healthcare*. Santiago de Compostela, Spain: 24th European Conference on Artificial Intelligence, 2020.

1.4 Estrutura da Dissertação

Os demais Capítulos desta Dissertação estão organizados conforme citado a seguir.

No Capítulo 2 são apresentados um estudo acerca do número de rebanhos de caprinos no país, os conceitos básicos relacionados a produção de couro caprino, o processo de qualificação das peças de couro *wet blue* e, por fim, os trabalhos que servem como base para a construção desta Dissertação.

No Capítulo 3 é apresentada a metodologia utilizada, ou seja, desde o treinamento vivenciado na empresa CV Couros e Peles Ltda, com o intuito de entender todo o processo de beneficiamento do couro caprino, até a elaboração do sistema de apoio à decisão para a qualificação das peças de couro caprino no estágio *wet blue*.

No Capítulo 4 são expostos os resultados obtidos a partir dos estudos e implementações explanados no capítulo anterior.

No Capítulo 5 são apresentadas as conclusões e as propostas para trabalhos futuros.

2 Revisão bibliográfica

Diversos estudos apontam o Brasil como um dos potenciais polos produtores da cadeia de produtos relacionados a couro, dada a existência de recursos naturais, como grande rebanho e mão-de-obra de custo baixo (SANTOS et al., 2018). No entanto, este potencial vem sendo reduzido devido a diversos fatores que permeiam desde a qualidade das peças produzidas até o uso de materiais sintéticos, que possuem aparência e consistência similar ao couro.

Com relação a baixa qualidade das peças produzidas, boa parte se deve ao formato rústico de criação dos animais, processamento das peças de couro e uma discriminação do nível de qualidade em formato artesanal e subjetivo.

Quanto ao uso de materiais sintéticos, no Brasil foi elaborada uma lei que proíbe o uso de expressões como couro sintético ou couro ecológico, a Lei 4.888, na qual proíbe a utilização do termo couro em produtos que não tenham sido obtidos exclusivamente de pele animal. A infração desta constitui-se crime de concorrência desleal.

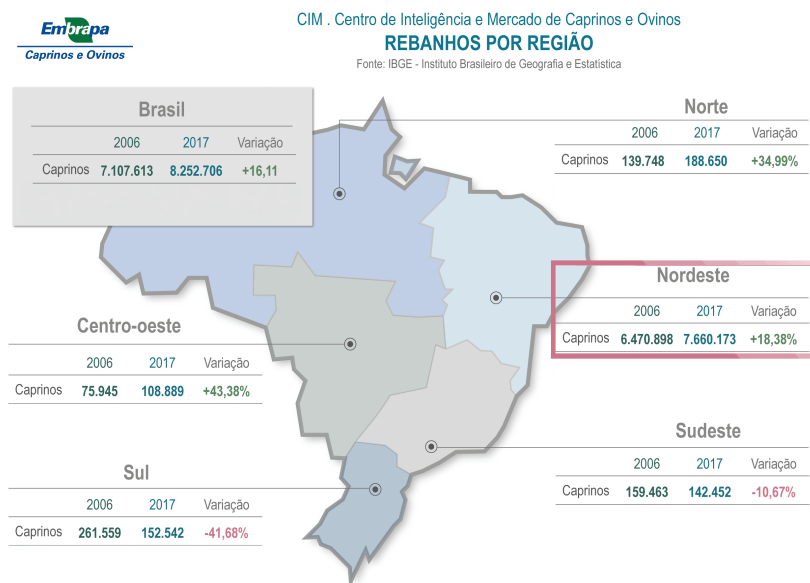
Deste modo, nota-se que há uma colaboração da União com este setor. Em contrapartida, existe a necessidade da inserção de metodologias que colaborem para a melhoria da qualidade das peças de couro produzidas e vendidas pelos curtumes. Assim, neste Capítulo, visando explanar a produção de couro no país, são apresentados conceitos a cerca da produção de caprinos nas regiões brasileiras, o processo de elaboração do couro, a definição do nível de qualidade das peças produzidas e, por fim, visando colaborar com este setor, os trabalhos existentes na literatura que são referenciais teóricos para a construção do programa computacional de apoio à decisão na qualificação das peças de couro caprino.

2.1 A produção de caprinos no Brasil

No último censo relacionado a rebanhos por região e efetuado pela Embrapa (2018), tendo como base a data deste trabalho, pode-se observar um crescimento no quantitativo de rebanhos de caprinos no país de aproximadamente 16,11 % (Figura 1).

Nota-se um maior crescimento da população de caprinos na região Centro-oeste, a qual possui uma variação de + 43,38 % seguida pela região Norte com + 34,99 %. No entanto, apesar do crescimento apresentado, a região Nordeste destaca-se por sua quantidade total de caprinos que, apesar de apresentar uma variação de apenas + 18,38 %, possui cerca de 92,82 % do total desses animais no país.

Figura 1 – Quantidade de rebanhos por região, valores referentes a 2006 e 2017.



Fonte: Adaptado de Embrapa (2018).

Percebe-se ainda um decréscimo de - 41,68 % na região Sul e - 10,67 % na região Sudeste, tais valores podem ser evidenciados devido a essas regiões concentrarem 91 % dos municípios mais desenvolvidos do país, locais onde a pecuária existe em menor escala.

Um dos fatores que contribuem para a criação de caprinos no Nordeste é citado por Leal, Rocha e Junior (2014), os quais explanam que, por ser de fácil adaptação ao clima semiárido, a espécie caprina se torna excelente opção para a pecuária no bioma caatinga.

A partir desses animais é possível obter diversos produtos como carne, leite, couro, sendo este último o de maior importância neste trabalho e utilizado para confecção de sapatos, cintos, bolsas, roupas, bumbos, tapetes, artefatos de montarias, entre outros.

No entanto, para gerar produtos de couro de qualidade deve-se ter uma preocupação inicial durante a criação do animal e onde ele transita. Deste modo, devido a diversos tipos de espinhos existentes no bioma caatinga e o uso de cercas de arame farpado, na maioria das vezes, por pequenos produtores rurais, há uma redução significativa na qualidade das peças advindas da região citada.

De acordo com Banco do Nordeste (2012), a má qualidade da pele é ponto crítico para pequenos produtores. Os defeitos nas peles não ocorrem apenas devido ao manejo, transporte e abate, mas na falta dos devidos cuidados na criação e esfolagem do animal. Desse modo, pequenos produtores respondem por importante parcela na depreciação da pele. A transferência de tecnologias simples, como o uso de faca rombuda ou simplesmente a conservação da pele na forma salgada já seria suficiente para melhorar a qualidade e a remuneração do produtor. O processo de salga já evitaria os danos do processo de espichamento e os custos adicionais de reidratação na indústria.

Ainda sobre a qualidade das peles, de acordo com a Embrapa (2002), entre outros fatores, a baixa qualidade torna o Brasil pouco competitivo no mercado internacional, além de apresentar dificuldades para suprir a atual demanda interna, tendo que recorrer a contínuas importações de peles para a manutenção do seu parque industrial.

Apesar das dificuldades encontradas, ainda é possível encontrar um quantitativo relevante de peles de qualidade, o que mantém este setor ainda firme no mercado. Tendo em vista o exposto, a seguir são apresentados os processos realizados pelos curtumes com o intuito de transformá-las em couro *wet blue*.

2.1.1 Processo de Beneficiamento Primário do Couro Caprino

O processo de beneficiamento do couro caprino é iniciado após o abate do animal. Após este último citado, é dado início ao processo de esfolagem, que consiste na retirada cuidadosa da pele com uso de uma faca, e salga, com o intuito de impedir a deterioração do couro.

Após a retirada e salga os produtores rurais recolhem as peles extraídas e as levam para feiras das cidades com o intuito de vendê-las a atravessadores, que repassam-nas para os curtumes.

Nos curtumes, as peles são aparadas com o intuito de remover partes desnecessárias como patas, rabo, pescoço, etc, e, conforme apresentada na Figura 2, são separadas por tamanho com o intuito de formar lotes uniformes.

Figura 2 – Separação das peles por tamanho.



Fonte: Cobrasil (2017).

Após a separação das peles por tamanho é iniciado o processo de *ribeira*, que consiste em um conjunto de operações e processos que preparam a pele para o curtimento. Dentre eles, destacam-se o *caleiro*, que consiste na depilação da pele, e o *descarne*, que possui o intuito de retirar o restante de gordura e carne da parte interna da pele.

O curtimento consiste em inserir as peles em *fulões* (Figura 3), nos quais são processadas com produtos químicos com o intuito de deixá-las prontas para o armazenamento. Este processo também visa torná-las mais resistentes, elásticas e duráveis, após este, as peças são denominadas como couro.

Figura 3 – Fulões.



Fonte: Adaptada de Cobrasil (2017).

De acordo com Tarantola (2014), o curtimento serve basicamente para mumificar uma pele e estabilizar o material resultante, para que não endureça demais nem apodreça. A atividade envolve preparar a pele, raspando-a para retirar qualquer carne, gordura ou pelo. Opcionalmente, também se aplica pasta de cal, branqueador ou produtos para matar as bactérias na pele.

Salienta-se que o curtimento pode ocorrer de forma vegetal ou mineral. No primeiro caso, geram couros mais pesados, que são utilizados em solas de calçados e o segundo, é o mais comum, destaca-se pela utilização de sais de crômio.

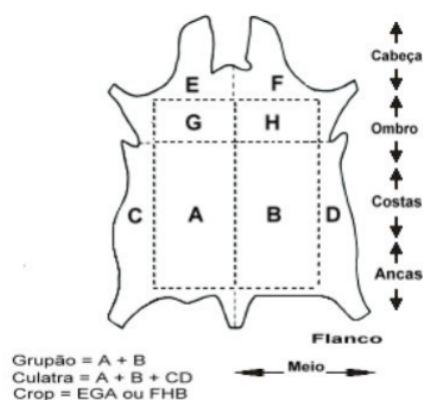
Após a retirada das peles dos fulões, as mesmas são denominadas como *wet blue*. Deste modo, é possível armazenar as peças de couro conforme sua qualificação (nível de defeitos) e vendê-las para outros curtumes ou indústrias de acabamento.

2.2 Qualificação do Couro

Conforme *Leather Dictionary* (2007), produtos com maior qualidade geralmente passam por extensos testes para garanti-la. Desse modo, fabricantes que evidentemente se esforçam para produzir um bom produto a partir do couro, geralmente obtêm um melhor nível qualidade. No entanto, nem todas as peças possuem a mesma qualidade, deste modo, há a necessidade de separá-las por nível.

Santos Filho (2013) esclarece que a separação das peças de couro em função da sua qualidade é realizada através de inspeção visual por um empregado do curtume, chamado de classificador. A classificação depende, além da quantidade de incidências de falhas no couro, da localização destes problemas. Assim, é de fundamental importância que o profissional conheça as regiões da pele para que possa ponderar as falhas de acordo com a sua localização (Figura 4).

Figura 4 – Representação pictográfica de regiões da pele de caprinos.



Fonte: Santos Filho (2013).

Uma peça de couro, após extraída do animal, divide-se em 8 áreas (A, B, C, D, E, F, G e H). A zona central da pele (A, B, G e H) contém regiões mais favoráveis para a comercialização do couro. Assim, falhas encontradas nesta localização tendem a reduzir mais a qualidade das peças de couro que nas demais regiões existentes.

No tocante às falhas, podem ocorrer desde a esfolação, ou seja, remoção da pele do animal, na qual poderá gerar cortes de faca, buracos e etc, até o processo de conservação, com a quantidade inadequada de conservante para manter uma boa qualidade das peças por um determinado período de tempo.

Salienta-se também, que algumas falhas, tais como picada de inseto, corte de arame, cicatrizes, entre outros, ocorrem, na maioria dos casos, enquanto o animal ainda não foi abatido. Isto posto, o fator criação do animal é de suma importância para a obtenção de uma pele de qualidade.

Após agrupar as peças de couro por nível de qualidade, os curtumes utilizam sete valores, no qual, o primeiro valor (1) representa peças que praticamente não apresentam defeitos visuais e à medida que este valor cresce, aumentam-se a quantidade de problemas encontrados, fazendo com que haja uma redução do valor dessas.

Com as peças agrupadas, as mesmas são vendidas para indústrias de acabamento onde ocorrem procedimentos complementares como: tingimento, engraxe, recorte, prensa-gem e medição, com o objetivo de obter as características finais.

Como citado, pretende-se construir um programa computacional que auxilie os profissionais dos curtumes no processo de qualificação das peças de couro. Deste modo, a seguir são apresentados os trabalhos base para a elaboração dessa proposta.

2.3 Trabalhos Relacionados

Amorim et al. (2010) propõem um trabalho de classificação de falhas em peças de couro de cabra em dois tipos de imagens: couro cru, ou seja, não processado, e *wet blue*, processado pelo curtume. Para a aquisição de atributos são utilizados *Gray-Level Co-Occurrence Matrix* (GLCM), filtros Gabor e dois espaços de cores.

A partir dos atributos adquiridos, são realizados três experimentos. O primeiro consiste em classificar os dados brutos, sem pré-processamento dos mesmos, através dos classificadores *K-Nearest Neighbors* (KNN), Bayes e *Support Vector Machine* (SVM), no qual, destaca-se a aplicação do classificador KNN, com taxa de precisão média de 95,9 % para peles de couro cru e 93,76 %, para *wet blue*.

No segundo experimento, são utilizadas técnicas de redução de atributos como *FisherFace*, *Chen Linear Discriminant Analysis* (CLDA), *Direct LDA* (DLDA), *Yang LDA* (YLDA) e *Kernel LDA* (KLDA) e, em seguida, os dados resultantes são novamente classificados com os mesmos classificadores do primeiro experimento. Nesta etapa, a melhor combinação é obtida pelo CLDA e KNN, obtendo uma taxa de precisão média de 92,2 % para peças de couro cru e 90,3 %, para *wet blue*.

O terceiro experimento consiste em detectar as falhas, informar a sua localização e comparar com as informadas pelo profissional especialista. Em uma análise visual, como resultado, o sistema encontrou diversos falsos positivos. Amorim et al. (2010) relatam que uma das razões para que ocorresse tal problema é a qualidade do conjunto de treinamento. Portanto, não possui amostras suficientes para discriminar todas as falhas possíveis.

Santos Filho (2013) propõe um modelo de classificação de amostras de peles no estágio *wet blue*. O conjunto de imagens se divide em sete tipos de classes que foram subdivididas em Classe Superior (classes 1 a 5) e Classe Inferior (classes 6 e 7). O trabalho realizado consiste em converter as imagens obtidas (3264 x 2448 *pixels*) para níveis de cinza e, posteriormente, redimensioná-las para tamanhos de 40, 60, 80 e 100 *pixels*. Em seguida é realizada a extração de características a partir dos métodos *Column-Variance* (VAR), *Haar wavelet transform* (HAAR), *Non-Negative Matrix Factorization* (NMF), *Principal Component Analysis* (PCA) e *Gray Level Co-occurrence Matrices* (GLCM).

Após adquiridos os vetores de características, é feita a classificação a partir dos classificadores *Least Squares* (LS), *Extreme Learning Machine* (ELM), *Regularized Linear Gaussian Classifier* (RLGC) e *Support Vector Machine* (SVM). No trabalho, consta ainda a elaboração da opção de rejeição com o intuito de aumentar a confiabilidade dos classificadores. Como melhores resultados do trabalho proposto, pode-se citar as taxas de acerto médio de 85,56 %, sem a opção de rejeição, através do extrator VAR e classificador ELM e, com opção de rejeição, 97,32 %, com PCA e SVM.

Aquino (2017) propõe a classificação de amostras de couro caprino a partir da

utilização de três variações do método *Local Binary Pattern* (LBP): LBP uniforme, LBP invariante à rotação e LBP uniforme e invariante à rotação. Para a classificação, são utilizados os classificadores *K-Nearest Neighbors* (KNN) e *Support Vector Machine* (SVM).

Após a análise da classificação a partir de cada extrator, é proposta também a extração de características com multi-resolução, ou seja, da combinação de dois ou três operadores LBP. Ao final do trabalho, percebe-se que a melhor acurácia média é de 86,14 %, obtida a partir da combinação de três operadores LBP (1-Uniforme, 2-Invariante à Rotação e 3-Uniforme e Invariante à Rotação) e utilização do classificador SVM.

Renato Pereira (2018), por sua vez, inicia seu trabalho a partir de quinze técnicas de segmentação (1- *Back Projection*, 2- *Black Hat*, 3- *Canny*, 4- Clusterização, 5- Casca convexa, 6- *Finding Contours*, 7- Transformada de Hough para círculos, 8- Transformada de Hough para linhas, 9- Momentos de imagem, 10- Transformada de Laplace, 11- Crescimento de Regiões, 12- Filtro de Desvio Padrão, 13- Sobel, 14- Limiarização de Otsu (1979) e 15- *Tophat*) a fim de detectar falhas nas amostras de couro caprino. Estas não mostraram bons resultados, visto que a melhor técnica foi a utilização do detector *Canny* que encontrou apenas 50 % das falhas.

Partindo para outra metodologia, Renato Pereira (2018) propõe a criação de janelas de falhas. Nesta, é elaborado um arquivo contendo a localização de regiões normais e sete tipos de falhas, após obtidas todas as localizações, são criadas janelas de 51, 101, 151, 201, 251 e 301 *pixels*. A partir dessas janelas, são elaborados os conjuntos de dados com a utilização dos métodos *Gray-Level Co-Occurrence Matrix* (GLCM), *Local Binary Pattern* (LBP), Momentos, *Structural Co-Occurrence Matrix* (SCM) e *Pixel Intensity Analyzer* (PIA). Após obtidos os dados, provindos das janelas elaboradas, são implementadas as classificações de falhas, a partir dos classificadores *K-Nearest Neighbors* (KNN), Bayes, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM).

Após o treinamento, o sistema analisa uma peça inteira de couro caprino, divide-a em janelas de igual tamanho ao que foi construído no conjunto de dados e segmenta as regiões de falhas. Com base nessas segmentações é elaborado um novo conjunto de dados a partir das regiões livres de falha, cujo o intuito é qualificar as peles entre os sete tipos de qualidade utilizados nos curtumes.

Uma problemática também analisada é o tempo de execução. Desta forma, em termos de melhor resultado a cerca da classificação das janelas de falhas, a melhor acurácia é obtida a partir de janelas de 51 x 51 *pixels*, extrator LBP e classificador MLP, com acurácia média de 90,25 %, porém, em termos de menor tempo de execução, o melhor resultado é adquirido utilizando a configuração de 51 x 51 *pixels*, extrator PIA e classificador MLP com acurácia de 87,95 %.

Quanto à qualificação das peles, os melhores resultados são obtidos a partir do

classificador SVM com acurácia média de 86,86 %.

Renato Pereira (2018), ao verificar que um profissional, especialista em qualificação de couros, leva em torno de 10 a 30 segundos para avaliar uma peça inteira, explana como melhor topologia no trabalho a classificação das janelas de falha a partir do tamanho 51 x 51 *pixels*, o extrator PIA, o classificador MLP para a classificação das janelas de falha e o SVM, para a qualificação das peças, o que resulta em uma acurácia média de 82,86 % em um tempo médio de 20 segundos.

Liong et al. (2019) propõem um sistema de detecção e segmentação de falhas do tipo picada de carrapato em amostras de couro bovino. Segundo os autores, a proposta por trabalhar com este tipo de defeito ocorre, pois, tal problema é frequentemente negligenciado por inspeção humana.

Nos procedimentos descritos no trabalho, são utilizados uma luz *Light-Emitting Diode* (LED), uma câmera de 24,2 *megapixels* com resolução espacial de 2400 x 1600 *pixels*, um braço robótico de mesa modelo DRV70L de seis eixos da empresa Delta Electronics, Inc. e utilização do método de aprendizado profundo *Mask Region-based Convolutional Neural Network* (Mask R-CNN).

A metodologia consiste em inserir amostras de couro manualmente em uma mesa na qual o braço robótico se movimenta em formato linear, de cima para baixo, analisando as peças e, a partir do uso da câmera, digitalizando a imagem a que foi visualizada. Com isso, a detecção de regiões consiste em salvar as coordenadas X e Y da região identificada como falha para posterior uso. Com o processo de classificação concluído, o braço robótico percorre novamente toda a peça de couro com o intuito de, a partir de um giz inserido em sua ponta, destacar as regiões anteriormente detectadas como falha. No experimento são utilizadas 84 imagens para treinamento e 500 imagens para testes, obtendo uma precisão de 70,35 %.

Diante do exposto, baseado nos trabalhos relacionados, a partir das técnicas que permitiram obter os melhores resultados e apresentando novas propostas, a seguir é apresentada a Metodologia.

3 Metodologia

Neste Capítulo, conforme apresentado na Figura 5, são apresentados os procedimentos metodológicos utilizados para a elaboração do sistema de apoio à qualificação de peças de couro caprino.

Figura 5 – Procedimentos metodológicos realizados para a elaboração do sistema



Fonte: Autor.

Conforme ilustrado na Figura 5, a metodologia consiste em quinze procedimentos que constituem o treinamento executado na empresa CV Couros e Peles Ltda, para a compreensão acerca do processo de qualificação de peles em um curtume, até os procedimentos implementados que compõem o programa computacional para qualificação das peças de couro caprino.

A seguir, como forma de obter maiores conhecimentos relacionados à detecção de falhas e definição do nível de qualidade das peças de couro caprino, é apresentado o primeiro procedimento metodológico, o treinamento efetuado na empresa CV Couros e Peles Ltda.

3.1 Treinamento na Empresa CV Couros e Peles Ltda

O primeiro estágio do desenvolvimento do sistema é iniciado a partir da obtenção de conhecimentos acerca do processo de qualificação das peças de couro. Esse, como citado, ocorreu na empresa CV Couros e Peles Ltda. Na Figura 6 é ilustrado o ambiente de trabalho

no qual ocorre o treinamento ministrado por três especialistas em classificação. Quanto aos profissionais envolvidos, visando a não identificação dos mesmos são denominados nesta dissertação como Especialista 1, Especialista 2 e Especialista 3.

Figura 6 – Imagens capturadas durante o treinamento na empresa CV Couros e Peles Ltda.



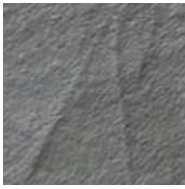
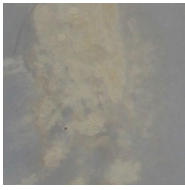
(a) *Processo de qualificação visual*

(b) *Peles qualificadas*

Fonte: Autor.

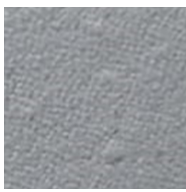
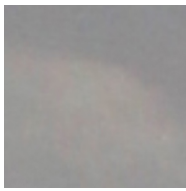
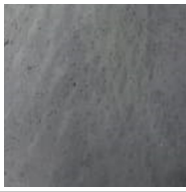
De acordo com o Especialista 1, a qualificação da pele se dá através de suas falhas, seus tipos e as suas localizações (mais ao centro, nas extremidades, etc). Nesse sentido, diversos tipos de falhas podem ser encontrados em peças de couro caprino. Nas Tabelas 1, 2 e 3 são apresentados os principais tipos detectados durante o treinamento realizado e uma breve descrição de cada.

Tabela 1 – Principais falhas encontradas em peças de couro caprino.

Imagem	Falha	Descrição
	Risco de Arame	Cortes causados em cercas de arame farpado.
	Corte de faca	Cortes ocasionados a partir do mau uso da faca durante a extração da pele do animal sem causar buracos na mesma.
	Má conservação	Manchas causadas por quantidade inadequada de sal ou demais compostos utilizados na conservação da pele.

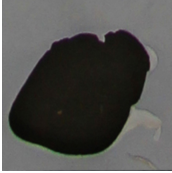
Fonte: Autor

Tabela 2 – Principais falhas encontradas em peças de couro caprino - continuação.

Imagem	Falha	Descrição
	Sinais	Mancha pigmentada de forma e tamanho indefinidos com coloração escura.
	Bexiga	Aparência de bolhas oriundas de doença de pele.
	Sarna	Manchas na cor branca ocasionadas por ácaros.
	Picada de inseto	Pequenos furos ocasionados por insetos.
	Cicatriz	Área de tecido fibroso, resultado da regeneração tecidual de uma lesão. Pode apresentar-se de maneira atrófica, hipertrófica e normotrófica, em relação à textura. Possui formato indefinido.
	Rufa	Marca ocasionada por pelo encravado, de característica pontiforme e esbranquiçada.
	Gordura Vegetal	Manchas ocasionadas por excesso de gordura vegetal. É caracterizada pela aparência estriada esbranquiçada ou de coloração escura.

Fonte: Autor

Tabela 3 – Principais falhas encontradas em peças de couro caprino - continuação.

Imagem	Falha	Descrição
	Buraco	Buracos causados pelo mau uso da faca durante a extração da pele.
	Conservação Forte	Manchas com tons vermelho ou laranja causadas por quantidade inadequada de sal ou demais compostos utilizados na conservação da pele.

Fonte: Autor

Durante o treinamento evidenciou-se que os problemas mais graves são provindos do corte de faca durante a esfolagem e da má conservação das peles. O motivo da má qualidade das peles é explicitado na fala, reproduzida a seguir, pelo especialista 3, cujas credenciais são 20 anos de experiência.

"[...] isso ocorre devido a desvalorização da matéria-prima, abaixo do valor comercial. Em alguns casos, as pessoas chegam a enterrar as peles pois não vale a pena vender"
(Especialista 3)

Ainda sobre as falhas, é válido salientar que, durante a classificação visual, um sinal pequeno pode ser confundido com picada de mosquito, bem como, sinais grandes possuem tonalidade escura e assemelham-se a buracos. Esses ocorrem devido ao mau cuidado durante a extração da pele do animal. Assim, em grande parte dos casos, esta falha é proveniente do corte de faca, porém, estas não possuem semelhanças.

Outro exemplo de dificuldade na análise visual é decorrente do mal armazenamento. Um empilhamento inadequado pode gerar dobraduras. Essas geram marcas semelhantes a cortes de faca ou cicatrizes. Tendo em vista tal problema, os especialistas viram a pele para o lado carnal e analisam se existe falha ou não (Figura 7).

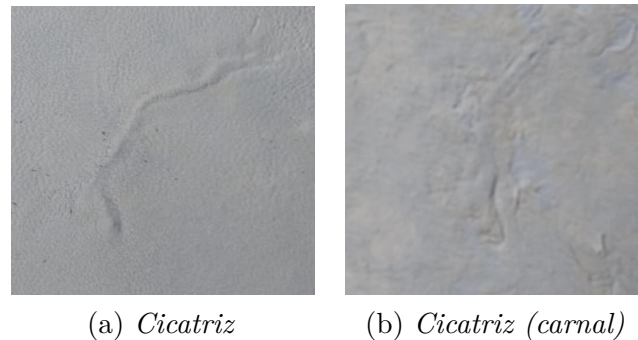
Figura 7 – Peça de couro no lado carnal. Em destaque a falha corte leve de faca.



Fonte: Autor.

Como citado, além do corte de faca, outra falha na qual os especialistas costumam verificar o lado carnal e que também são provenientes do mal empilhamento, é a cicatriz, demonstrada na Figura 8. Conforme apresentado, ao virar a pele para o lado carnal, torna-se possível identificar a falha com mais exatidão.

Figura 8 – Exemplo de falha relacionada à cicatriz.



(a) *Cicatriz*

(b) *Cicatriz (carnal)*

Fonte: Autor.

Problemas provindos da má conservação podem possuir aparências distintas. Desta forma, podem apresentar regiões secas, esbranquiçadas ou avermelhadas.

Verifica-se também que um dos tipos de falha bastante complexo de se analisar é a sarna, pois, conforme os profissionais da empresa, estas, em tons mais fracos, não são identificadas como defeitos nas peças de couros.

3.2 Aquisição do Conjunto de Amostras de Couro Caprino

Após o processo de obtenção de conhecimento acerca do processamento das peles e dos defeitos existentes nelas, os profissionais do curtume disponibilizaram 312 amostras, divididas conforme indicado na Tabela 4. Estas foram anteriormente qualificadas pelo Especialista 1 e já se encontravam em estoque na empresa.

Tabela 4 – Quantidade de peles recebidas.

Qualidade	1	2	3	4	5	6	7
Quantidade	21	44	48	50	49	50	50

Com o intuito de analisar o processo visual de qualificação das peles entre diferentes especialistas de uma mesma empresa, as peças recebidas são novamente qualificadas.

Para garantir que o segundo especialista não possua qualquer tipo de referência e preservar a qualificação previamente realizada, as amostras disponibilizadas são etiquetadas.

Conforme apresentadas na Figura 9, as etiquetas possuem 6 caracteres aleatórios (4 números e 2 letras), definidos por um algoritmo de geração de caracteres aleatórios de

modo a não produzir códigos repetitivos e que não induzisse o profissional a qualquer classe.

Figura 9 – Etiquetas inseridas nas peles.



Fonte: Autor.

O processo de inserção das etiquetas é trabalhoso, pois, é feito manualmente com o cuidado de não causar danos às peças. São feitos furos com agulhas e as etiquetas amarradas através deles com linha nylon de 0,30 mm. O procedimento é repetido em cada uma das peças de couro disponibilizadas, sendo que as mesmas são, conforme apresentadas na Figura 10, misturadas para não gerar qualquer vício durante a nova qualificação.

Figura 10 – Parte das amostras prontas para requalificação.



Fonte: Autor.

Após etiquetar todas as peles, ocorre a requalificação pelo especialista 2, apresentada na Figura 11. O profissional recebe as peles totalmente misturadas e qualifica-as. É importante salientar que ele desconhece qualquer padrão contido nas etiquetas. Desta forma, não há a possibilidade de ocorrer vícios durante o processo.

Figura 11 – Especialista requalificando as peças de couro.



Fonte: Autor.

Em sequência à requalificação do especialista 2, é gerado um arquivo (vide Apêndice A) contendo as informações referentes às qualificações dos dois profissionais. O principal intuito é de selecionar amostras de cada categoria com maior grau de confiança, ou seja, das amostras que possuem a mesma categorização por parte dos dois avaliadores.

3.2.1 Análise da Classificação Visual dos Especialistas

Um aspecto importante é avaliar o grau de concordância entre os dois avaliadores. Com isso pode-se inferir o grau de subjetividade e dificuldade na tarefa que se propõe a realizar.

Ao analisar a Tabela 5 e considerando que as células em destaque, na cor cinza, diagonal principal, representam a quantidade de peles que os dois especialistas qualificaram da mesma forma. Percebe-se que existem muitas inconsistências em relação às qualificações. Isto demonstra o quão complexa é a solução para esta problemática.

Tabela 5 – Matriz de Confusão com o rótulo dos especialistas.

Especialista 1 \ Especialista 2	Especialista 2						
	1	2	3	4	5	6	7
1	8	6	2	2	3	0	0
2	9	17	7	5	5	1	0
3	2	4	11	1	16	13	1
4	0	2	3	20	18	7	0
5	0	0	0	1	21	21	6
6	0	0	0	1	3	9	37
7	0	0	0	0	0	5	45

Desta forma, nota-se um índice de concordância de 42 %. O resultado parece bastante consistente, dado que no trabalho de Renato Pereira (2018) a concordância é de 41,37 %.

Após a análise do grau de semelhança na qualificação visual das peles, é dado início à criação do conjunto de imagens.

Para a aquisição das imagens de couro caprino, é utilizada uma mesa com fundo preto, iluminação própria composta por duas lâmpadas fluorescentes de 40 W dispostas de tal forma a não gerar sombras e um suporte para câmera situado a 1,5 m acima da superfície da mesa. Isto produz melhores resultados na obtenção dos dados pós-coleta das imagens. Tal estrutura foi desenvolvida por Santos Filho (2013) e é apresentada na Figura 12.

Figura 12 – Mesa para fotografia das amostras.



Fonte: Autor.

A Câmera fotográfica utilizada é da marca Canon, modelo T3 Rebel de 12.2 MP e zoom óptico de 55x, lente EF-S 18-55mm e *Joint Photographics Experts Group* (JPEG) como formato de arquivo utilizado. A mesma é conectada a um computador com *software Canon EOS Utility*[®].

O processo de captura das amostras de couro caprino ocorre inserindo as peças de couro na mesa citada anteriormente e capturando-as a partir da câmera fotográfica. Este é apresentado na Figura 13. Após a captura as imagens são salvas em um disco rígido e denominadas com o rótulo proveniente das etiquetas inseridas nas peças de couro.

Figura 13 – Captura das amostras de couro caprino disponibilizadas.



Fonte: Autor.

Após o processo citado anteriormente, de acordo com os rótulos salvos no relatório,

as amostras são analisadas e, caso possuam classificações idênticas pelos especialistas, são inseridas no conjunto de imagens.

3.3 Aquisição do Conjunto de Imagens de Falha

Como mencionado anteriormente, a qualificação do couro caprino consiste em verificar os tipos de falhas existentes, a quantidade e a localização. Desta forma, visando tornar o processo mais próximo possível da metodologia utilizada nos curtumes, é proposta uma detecção de falhas nas imagens capturadas com o intuito de identificar possíveis defeitos contidos em cada amostra de couro.

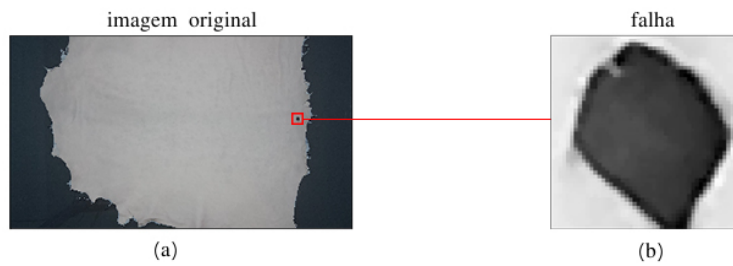
Ao analisar os diversos tipos de falhas, pode-se verificar semelhanças entre as mesmas e algumas divergências dentro da própria classificação. Sendo assim, com o auxílio dos especialistas em qualificação da empresa CV Couros e Peles Ltda, as falhas são agrupadas conforme ilustrado na Tabela 6.

Tabela 6 – Agrupamentos elaborados (falhas e regiões normais).

Rótulo	Classificação
0	Normal
1	Sinais leves ou picadas de mosquito
2	Sinais mais graves ou buracos
3	Sarna
4	Cortes leve de faca
5	Riscos de arame
6	Marcas de espinhos ou demais cicatrizes da vegetação
7	Bexigas
8	Cicatrizes
9	Gordura vegetal
10	Má conservação
11	Conservação forte

Para que se possa distinguir regiões de interesse de couro com defeitos dos isentos de defeitos, fazem-se necessárias amostras destas. Assim, com o auxílio de profissionais do curtume, as imagens capturadas são analisadas em um processo que consiste em abrir cada imagem em um *software* editor, disponibilizá-la para avaliação do especialista e, em seguida, catalogar a posição de cada região de interesse apontada por ele (b) e a sua designação (tipo da falha ou região normal). A Figura 14 (a) ilustra o descrito.

Figura 14 – Exemplo de localização de falha.



Fonte: Autor.

Após este processo, a localização, eixos X e Y da região apontada é catalogada em um arquivo, conforme apresentado na Tabela 7, bem como os atributos: número da imagem, nível de qualidade e a designação da falha de acordo com os agrupamentos elaborados (Tabela 6).

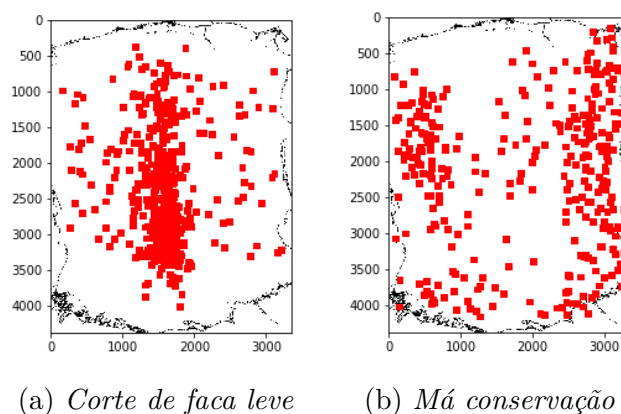
Tabela 7 – Exemplo de alguns dados contidos no relatório elaborado.

Imagem	Qualificação da pele	X	Y	Tipo de falha
1	7	823 px	3176 px	4
1	6	1029 px	1603 px	5
1	5	1704 px	1244 px	6

Após o processo de criação do relatório citado anteriormente, são obtidas 1669 regiões isentas de defeitos e 2250 com defeitos.

Em geral, os defeitos não apresentam um padrão de incidência na peça de couro. Entretanto, como ilustrado na Figura 15, o corte leve de faca tem uma tendência a ocorrer nas proximidades do dorso do animal (centro da peça) e a má conservação tem sua maior frequência nas bordas.

Figura 15 – Localização de regiões de falhas.



Fonte: Autor.

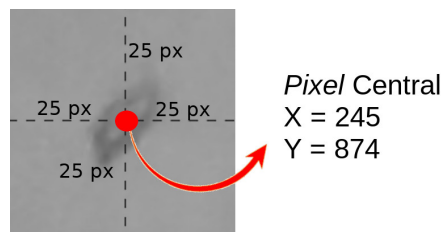
Os problemas evidenciados anteriormente possuem uma maior incidência nos locais apresentados pelo fato de que, no primeiro caso, há uma maior dificuldade na retirada da pele do animal em regiões próximas a sua coluna. No segundo caso, pelo fato de que as regiões mais próximas às bordas não serem tão aproveitadas como as regiões centrais, deste modo, havendo um menor cuidado com relação a estas áreas.

3.4 Aquisição dos Conjuntos de Dados para Detecção de Falhas

Após a conclusão do processo de obtenção das localizações de imagens com falhas e regiões normais, explanado na Sessão 3.3, é iniciado o processo de extração de atributos para a geração dos conjuntos de dados.

É necessário, inicialmente, estabelecer o tamanho da região de interesse de imagem para que sejam aplicados os métodos de extração de atributos. Como apresentado na Figura 16, a partir de um dado pixel, a região de interesse é definida como um quadrado baseado nas coordenadas adquiridas pelas áreas identificadas pelo especialista do curture e que foram anteriormente aplicadas no relatório apresentado na Seção 3.3.

Figura 16 – Exemplo de região de interesse de tamanho 51 x 51 *pixels*



Fonte: Autor.

Nesta etapa, são utilizados diversos tamanhos de regiões de interesse para a aquisição de características. Isto posto, são utilizados os tamanhos de 11 x 11, 21 x 21, 31 x 31, 41 x 41, 51 x 51, 75 x 75 e 101 x 101 *pixels*, resultando em diversos conjuntos de dados. Salienta-se que os valores superiores a 41 x 41 são adotados devido aos resultados apresentados na literatura e, os demais tamanhos, visando a busca de melhores resultados.

Após a aquisição das regiões de interesse, detectadas como normal ou falha, é iniciado o processo de extração de atributos com os extratores LBP e GLCM (apresentados no Anexo A.6).

Inicialmente é aplicado um filtro da mediana (Anexo A.2.2) com tamanho de *kernel* 3 x 3 em cada região de interesse e, em seguida é utilizado o extrator de características.

Para o extrator LBP, existem diversas formas de configuração dos parâmetros, conforme explicado no Anexo A. Como forma de obter os melhores resultados possíveis

com este extrator, são utilizados os mesmos valores propostos por Ojala, Pietikäinen e Mäenpää (2002), que são apresentados na Tabela 8.

Tabela 8 – Parâmetros utilizados no extrator LBP.

Ordem	Pontos	Raio
1	4	1,0
2	8	1,0
3	12	1,5
4	16	2,0
5	24	3,0

A partir das combinações entre os extratores de características, os parâmetros e as dimensões de regiões de interesse, são adquiridos sete conjuntos de dados a partir do extrator GLCM e mais sete a partir de cada combinação de parâmetros do LBP. Estes são apresentados conforme a Tabela 9.

Tabela 9 – Conjuntos de dados adquiridos a partir de regiões de falhas.

Extrator	Regiões de interesse	N. de Atributos	N. de Conjuntos
GLCM	11, 21, 31, 41, 51, 75 e 101 <i>pixels</i>	13	7
$LBP_{1,4}$		6	7
$LBP_{1,8}$		10	7
$LBP_{1,5,12}$		14	7
$LBP_{2,16}$		18	7
$LBP_{3,24}$		26	7
Total de conjuntos de dados adquiridos			42

Conforme apresentado, são utilizadas as dimensões de regiões de interesse iguais a 11 x 11, 21 x 21, 31 x 31, 41 x 41, 51 x 51, 75 x 75 e 101 x 101 *pixels* e 5 combinações dos parâmetros número de pontos e raio para o extrator LBP. A consequência dessas escolhas é a aquisição de 35 conjuntos de dados provindos do extrator LBP, ou seja, 5 conjuntos para cada dimensão escolhida, e 7 do GLCM, totalizando 42 conjuntos de dados.

3.5 Organização dos conjuntos de dados e utilização dos algoritmos de aprendizado de máquina

Como citado na Seção 3.3, são obtidas 1669 regiões isentas de falhas e 2250 com falha. Após gerados os conjuntos de dados, a partir dessas regiões e dos extratores LBP e GLCM, faz-se necessário utilizar um processo com o intuito de classificar áreas em amostras de couro caprino. Neste sentido, são utilizadas sete técnicas de aprendizado de máquina que, para o treinamento, os conjuntos de dados são divididos aleatoriamente em valores que representam 80 % da classe com menor quantidade de amostras, ou seja, 2670

amostras divididas igualmente entre normais e falhas, e o restante do universo de dados, isto é, 334 amostras normais e 915 falhas, é aplicado para o processo de teste.

Considerando-se que escalas de valores diferentes dos atributos nos conjuntos de dados podem dificultar os algoritmos de classificação, há a necessidade de que todos os dados sejam redimensionados para a mesma ordem de grandeza. Neste caso, aplica-se o processo de normalização *Z-score* aos conjuntos de dados elaborados, dados por

$$x' = \frac{x - \bar{x}}{\delta}, \quad (3.1)$$

pois considera-se x' como o valor normalizado, x o valor do atributo contido no conjunto de dados, $\bar{x}$ a média dos valores a serem analisados e δ , o seu desvio padrão.

Posterior a execução da normalização dos conjuntos de dados é iniciado o processo de aprendizado de máquina, para este são utilizadas as técnicas *Single Layer Perceptron*, *Logistic Regression*, *K-Nearest Neighbors* (KNN), classificador de Fisher, *Naïve Bayes*, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM). Para mais informações acerca destes algoritmos vide Anexo B.

O *Single Layer Perceptron* é inserido nesta Dissertação com o intuito de verificar a possibilidade de separabilidade linear entre as classes dos conjuntos de dados. O perceptron foi implementado utilizando taxas de aprendizado com valores de 0,01, 0,05, 0,1 e 0,5, com possibilidade de redução linear ou exponencial delas e função de ativação sinal.

Apesar do perceptron de única camada, possuir uma simples implementação, a sua principal desvantagem é não convergir se as classes não são perfeitamente linearmente separáveis. Deste modo, como forma de introduzir um algoritmo de simples implementação, e com melhor possibilidade de convergir para problemas não lineares, é utilizada a técnica de *Logistic Regression* a qual para o valor de C, ou seja, para o parâmetro de penalidade são adotados os valores 0,01, 0,1, 1,0 e 10,0.

A partir dos conjuntos de dados obtidos durante as extrações de características, descritos na Seção 3.4, notou-se aglomerações de amostras, o que torna um bom indicativo para o uso de uma das alternativas mais simples para classificação, o *K-Nearest Neighbors* (KNN). Este algoritmo explora a ideia de *lazy learning*, ou seja, não constrói seu aprendizado por meio de treinamento. Sua ideia principal é determinar a classe de uma amostra baseada nas amostras mais próximas dela. Em linhas gerais, este requer basicamente uma métrica de distância na qual foi utilizada a distância euclidiana e para o valor de K, que consiste na quantidade de vizinhos a serem analisados, foram utilizados os valores 3, 5, 7, 9 e 11.

O classificador de Fisher, baseado na regra de Bayes, é um método de classificação bem conhecido, amplamente utilizado em muitos campos, incluindo diagnóstico médico, robótica e visão computacional (GVARDINSKAS, 2017). Diferentemente do KNN, por mostrar-se bastante lento ao utilizar uma grande quantidade de amostras no conjunto

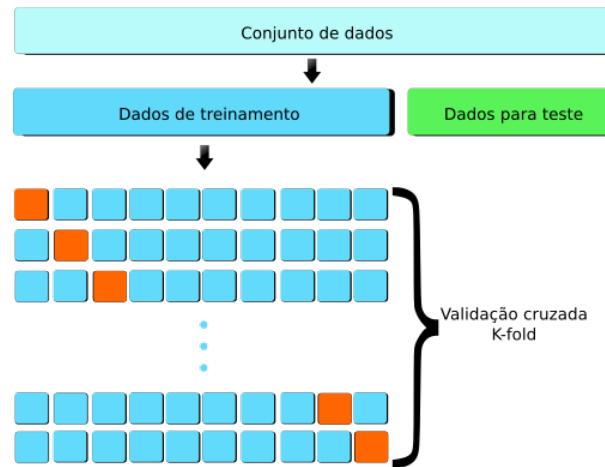
de dados, este é adaptável a diversos números de amostras e o seu treinamento pode ser realizado em tempo linear.

Visando a utilização de mais um algoritmo que faz previsões computando a probabilidade de uma determinada entrada pertencer a uma classe é implementado o classificador *Naïve Bayes*, este, diferente do classificador de Fisher, assume que cada elemento faz uma contribuição independente para a probabilidade atribuída a uma classe, ou seja, assume a independência condicional entre os atributos.

Silva, Spatti e Flauzino (2019) cita que as redes neurais são caracterizadas por possuir elevadas possibilidades de aplicações em diversos tipos de problemas relacionados as mais diferentes áreas do conhecimento, sendo também consideradas uma das arquiteturas mais versáteis quanto à aplicabilidade. Haykin (2001) apresenta que o *Multilayer Perceptron* (MLP) têm sido aplicado com sucesso para resolver diversos problemas. Este tipo de rede neural é caracterizada pela presença de vários perceptrons interligados de modo a possuir pelo menos uma camada intermediária, denominada camada oculta. Nesta Dissertação, este classificador é implementado a partir do uso de diversas taxas de aprendizado, que variam entre 0,1 a 0,5, com possibilidade de redução linear ou exponencial dessas, funções de ativação logística ou tangente hiperbólica e diversos números de neurônios na camada oculta, definidos empiricamente com valores de 2 a 100 neurônios.

Por fim, uma das técnicas de classificação e regressão mais utilizadas atualmente é o *Support Vector Machine* (SVM). Este algoritmo cria um hiperplano que separa amostras de diferentes tipos de classes e consegue trabalhar com eficiência na maioria dos problemas não lineares, fazendo uso de métodos de kernel. Na literatura pode-se encontrar diversos destes, sendo que nesta Dissertação são utilizados 4 tipos: linear, *Radial Basis Function* (RBF), polinomial e sigmoide. Para a variável C são adotados os valores 0,01, 0,1, 1,0 e 10,0.

Salienta-se que para todos os classificadores que necessitam de treinamento é utilizada a técnica de validação cruzada *K-Fold*, conforme mostrado na Figura 17. Essa técnica consiste em dividir o conjunto total de dados do treinamento em K subconjuntos de mesmo tamanho. Nesta Dissertação é adotada a validação cruzada pois, nas amostras com falhas, como apresentado na Seção 3.3, há 11 subconjuntos. Deste modo, com a utilização, visa-se obter um melhor processo de aprendizado. Salienta-se que para o parâmetro K é adotado um valor igual a 10.

Figura 17 – Exemplo de aplicação do método de validação cruzada *K-Fold*.

Para os classificadores citados é programada a execução de 2000 iterações. No entanto, se a taxa de erro atingir um valor menor ou igual a 0,00001, o treinamento é interrompido. A cada iteração dos algoritmos, é salva a matriz de pesos. Assim, é possível fazer uso dos melhores resultados obtidos dentro das execuções efetuadas.

O processo de escolha do melhor classificador é constituído de três passos consecutivos. O primeiro consiste em uma busca exaustiva em grade, ou seja, na utilização pré-definida de diversos hiperparâmetros para os processos de treinamento nos classificadores. O segundo, a partir do uso dos melhores hiperparâmetros encontrados, é executado a partir de pequenas alterações nos valores relacionados aos melhores hiperparâmetros obtidos e o terceiro, consiste na utilização dos melhores modelos da etapa anterior treinados 50 vezes, visando garantir a melhor escolha.

3.6 Métricas de avaliação dos classificadores propostos

Em problemas de classificação, diversas métricas podem ser utilizadas para mensurar cada modelo implementado.

Nesta Dissertação, além da matriz de confusão representada na Tabela 10, composta pelos valores relacionados a Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP) e Falso Negativo (FN), também são utilizadas as métricas denominadas acurácia, precisão, *recall* e *F1*, as quais são explanadas a seguir.

Tabela 10 – Matriz de Confusão

Classe correta	Classe predita		
		Normal	Falha
	Normal	Verdadeiro Positivo	Falso Negativo
	Falha	Falso Positivo	Verdadeiro Negativo

A acurácia é uma das métricas mais simples e indica o índice de classificação correta com base no total de amostras. Porém, nem sempre o uso dela como métrica de avaliação garante a escolha do melhor modelo a ser utilizado, pois, em conjuntos de dados desbalanceados, uma acurácia alta pode representar uma alta taxa de acerto em uma classe com mais amostras. Entretanto, pode ocorrer diversos erros em uma classe com poucas amostras, tornando a escolha de um modelo de classificação com base na acurácia uma escolha ruim. Esta métrica de avaliação pode ser representada conforme a Equação 3.2 dada a seguir.

$$acc = \frac{VP + VN}{VP + VN + FP + FN}. \quad (3.2)$$

A precisão é a razão entre o número de verdadeiros positivos (VP) e o total de observações positivas previstas pelo algoritmo (VP e FP), dada pela Equação 3.3 dada a seguir.

$$prec = \frac{VP}{VP + FP}. \quad (3.3)$$

Deste modo, pode-se afirmar que a precisão mensura a capacidade do classificador não rotular como positiva uma amostra negativa, ou seja, calcula a exatidão de quantos verdadeiros positivos são classificados corretamente.

O *Recall* ou Sensibilidade é a razão entre o número de verdadeiros positivos (VP) e todas as amostras da classe analisada. Assim, mensura a frequência com que o classificador classifica corretamente uma determinada classe. Esta métrica de avaliação pode ser representada pela Equação 3.4 dada a seguir.

$$rec = \frac{VP}{VP + FN}. \quad (3.4)$$

De acordo com Browniee (2020), muitas vezes a medida *recall* se torna conflitante com a precisão, de modo que ao aumentar o valor relacionado a uma, reduz-se o valor existente da outra. Deste modo, faz-se necessário encontrar um equilíbrio, que considere tanto os falsos positivos quanto os falsos negativos. Isto posto, conforme Shung (2018), a pontuação *F1* pode ser a melhor medida a ser utilizada de modo a buscar um equilíbrio entre a precisão e o *recall* que calculada a partir da média harmônica ponderada entre a precisão e o *recall*, representada pela Equação 3.5, dada a seguir.

$$f1 = 2 * \frac{prec * rec}{prec + rec}. \quad (3.5)$$

Como apresentado anteriormente, são utilizadas 4 métricas de avaliação para mensurar o nível de aprendizado dos classificadores. Nesta Dissertação, estas são empregadas

objetivando a melhor escolha nos processos de detecção de falhas e qualificação das peças de couro caprino.

3.7 Detecção de Falhas em Couro Caprino

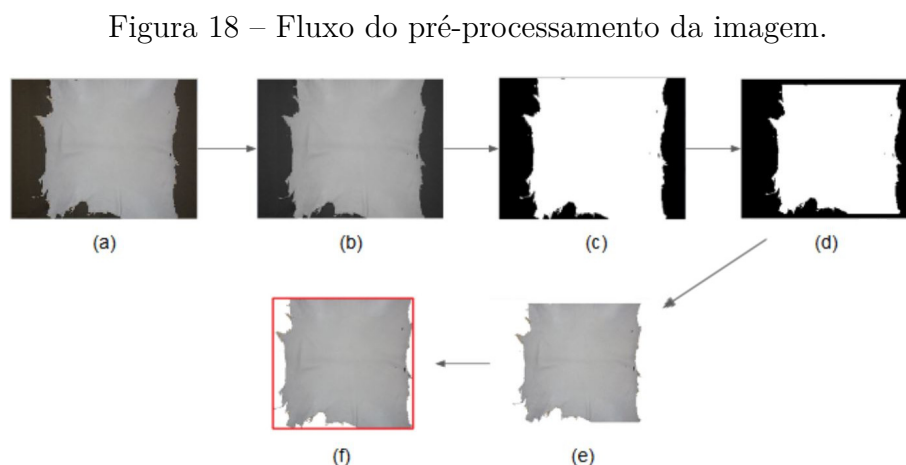
Com o intuito de analisar a pele inteira, após selecionados os melhores algoritmos e conjunto de dados, é iniciado o processo de localização de falhas em peças de couro caprino. Este processo é dividido em quatro etapas: pré-processamento, elaboração de regiões de interesse da imagem de couro caprino, classificação destes e execução do algoritmo de localização de falhas.

3.7.1 Pré-processamento que antecede a detecção de falhas

O pré-processamento consiste em remover o plano de fundo da imagem, restando apenas o objeto de interesse, ou seja, a peça de couro. Em vista disso, são implementadas e avaliadas 3 (três) técnicas de pré-processamento as quais, apenas a que obtém melhor resultado é utilizada. Essas técnicas são explanadas a seguir.

3.7.1.1 Pré-processamento baseado em crescimento de região

A primeira metodologia aplicada consiste na técnica utilizada por Renato Pereira (2018), representada na Figura 18. Nela o autor adquire a imagem original (a), transforma-a em tons de cinza (b) e em seguida aplica uma limiarização de Otsu (1979) (c).



Fonte: Adaptada de Renato Pereira (2018).

Com a imagem limiarizada, são inseridas 10 linhas na parte inferior e superior da imagem e 10 colunas nas partes laterais (d) com o intuito de garantir o crescimento de regiões atingindo todo o plano de fundo em peças de couro localizadas próximas as bordas das imagens.

Após a inserção de linhas e colunas é dado início ao crescimento de regiões (Anexo A.4) a partir do *pixel* semente localizado na posição $([0,0])$, ou seja, no canto superior esquerdo da imagem. A posição citada é a escolhida pois nela consta apenas o plano de fundo da imagem, área desnecessária para a análise da peça de couro caprino.

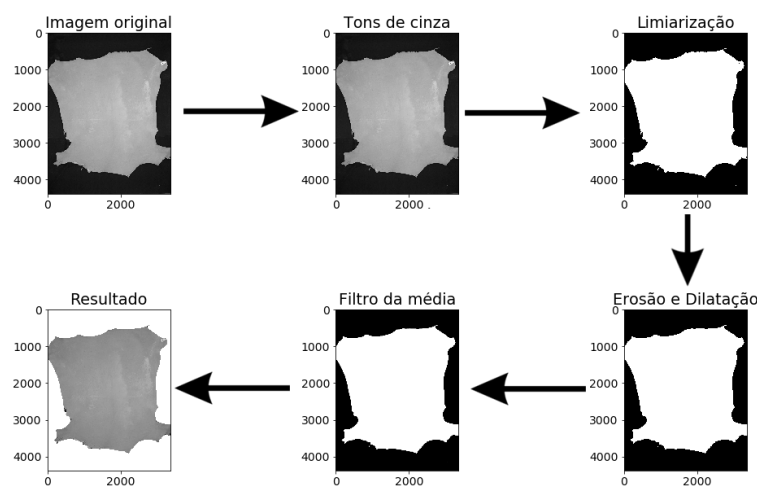
Com a técnica de crescimento de regiões executada é aplicado um processo iterativo *pixel a pixel*, no qual, se o *pixel* analisado for branco, a nova imagem recebe a cor do *pixel* respectivo da imagem original. Em caso contrário, a nova imagem recebe a cor branca (e), o resultado deste processo é a peça de couro caprino e um plano de fundo na cor branca.

Após a aplicação do método citado, é utilizada a técnica de casca convexa (f), que gera uma área convexa ao redor da peça de couro caprino. O resultado desta aplicação possibilita a exclusão de área não relevante para a solução do problema, ou seja, que não fazem parte da peça de couro caprino.

3.7.1.2 Pré-processamento baseado em Morfologia Matemática

Conforme apresentada na Figura 20, a segunda metodologia consiste em utilizar técnicas de operações morfológicas (vide Anexo A.5), precisamente execuções dos métodos de Erosão e Dilatação durante o pré-processamento das amostras de couro.

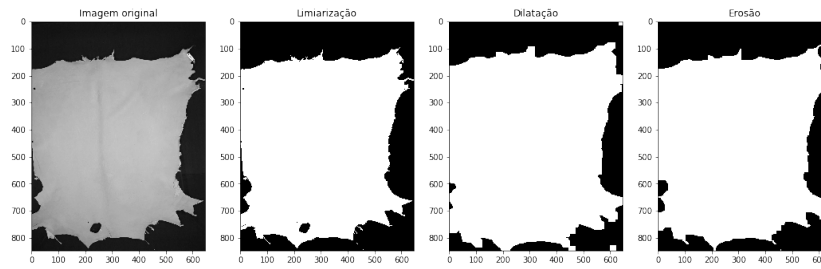
Figura 19 – Técnica de pré-processamento baseada em Erosão e Dilatação.



Fonte: Autor.

Semelhante ao início da metodologia anterior, a imagem original é transformada em tons de cinza e aplicada a limiarização de Otsu (1979) com o intuito de binarizar a imagem. Após os procedimentos citados é aplicada a técnica de dilatação, com o intuito de que falhas relacionadas a buracos sejam reconhecidas pelo programa computacional como parte da peça de couro e posteriormente é aplicado o método de erosão, visando remover eventuais ruídos que possam surgir.

Figura 20 – Aplicação de dilatação e erosão em uma imagem binarizada.



Fonte: Autor.

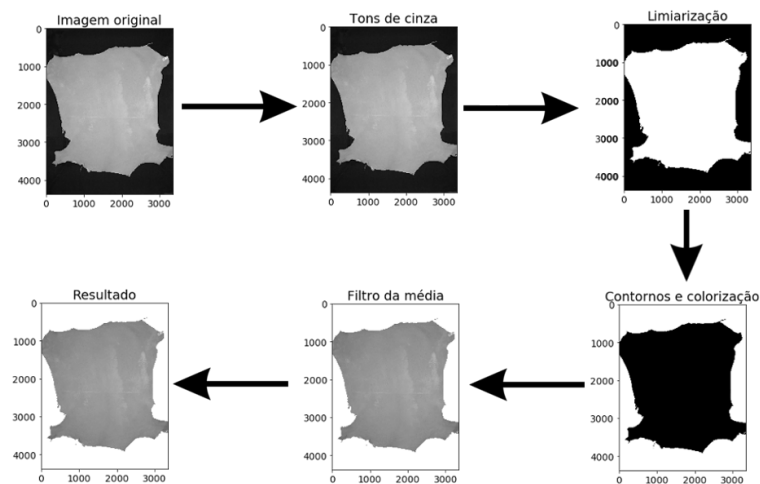
Após a aplicação dos métodos anteriormente citados, é aplicado na imagem resultante o filtro da média com dimensão 3×3 (Anexo A.2.1). Este filtro é aplicado com o intuito de remover algum ruído que não tenha sido removido com os procedimentos anteriores.

Dando sequência, semelhante à metodologia de pré-processamento anteriormente explanada, é implementado um processo *pixel a pixel* na imagem resultante substituindo a cor branca pela imagem original e a cor preta, por branca. O resultado dessa ação é a peça de couro caprino e um plano de fundo na cor branca.

3.7.1.3 Pré-processamento baseado em detecção de bordas

A terceira metodologia, apresentada na Figura 21, é semelhante a anterior. Desta forma, consiste em adquirir a imagem original, transformá-la em tons de cinza, limiarizá-la.

Figura 21 – Técnica de pré-processamento baseada na Detecção de Bordas.

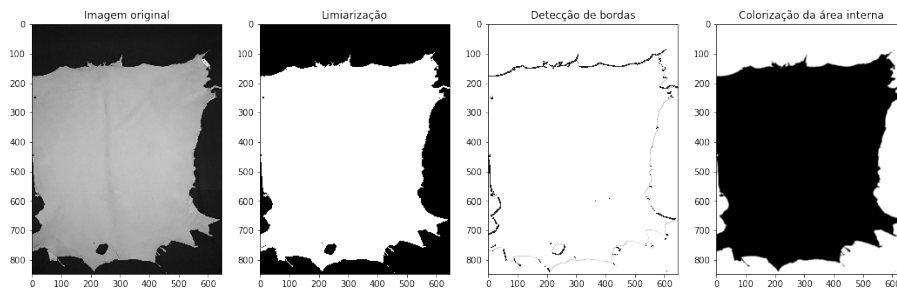


Fonte: Autor.

Após executar o processo de limiarização é empregada a técnica de filtro passa-alta Sobel (explanada no Anexo A.3.1), esta é aplicada com a combinação das máscaras vertical e horizontal. O objetivo da mesma é detectar as bordas da peça de couro caprino.

Com a obtenção dos contornos da peça de couro caprino, apresentados na Figura 22, toda a área interna do contorno mais externo, ou seja, da borda do objeto de interesse, recebe a cor preta. O resultado desta aplicação é o plano de fundo na cor branca e toda a peça de couro caprino na cor preta.

Figura 22 – Detecção do contorno e colorização da peça de couro caprino.



Fonte: Autor.

Em seguida, é iniciada a verificação e substituição em cada *pixel*. Esta consiste em verificar todos *pixels* da imagem, e caso possuam a cor preta deverão ser substituídos pela cor do *pixel* respectivo da imagem original.

Após os procedimentos descritos anteriormente, é aplicado um filtro da média, com máscara de tamanho igual a 3×3 , com o intuito de reduzir eventuais ruídos que possam surgir.

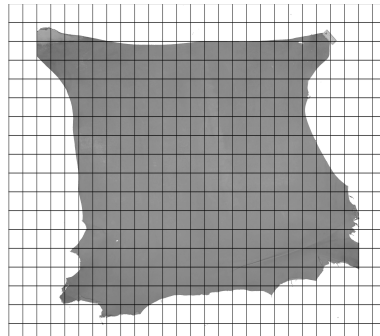
Como resultado desta metodologia, todo o plano de fundo possui a cor branca, restando em destaque, a peça de couro caprino.

3.7.2 Divisão em regiões de interesse

Nos curtumes, conforme citado na Seção 3.1, a qualificação do couro se dá através dos tipos de falhas e suas localizações. Desta forma, ao analisar a metodologia proposta por Renato Pereira (2018), torna-se claro que essa técnica é bastante próxima da utilizada pelos profissionais em uma inspeção visual, pois, baseia-se em detectar regiões de interesse de falha na peça de couro e destacar a sua localização.

Seguindo a mesma metodologia citada, conforme mostrado na Figura 23, a imagem de couro caprino pré-processada é fragmentada em diversos modos diferentes. Desta forma, são utilizadas as seguintes dimensões de regiões de interesse: 11×11 , 21×21 , 31×31 , 41×41 , 51×51 , 75×75 e 101×101 *pixels*.

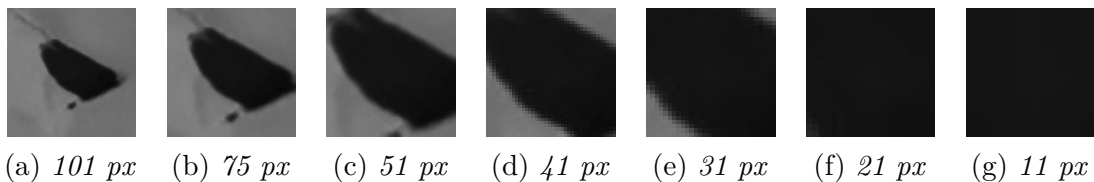
Figura 23 – Exemplo de divisão em regiões de interesse.



Fonte: Autor.

Conforme mostrado na Figura 24, dimensões diferentes de regiões de interesse representam formas distintas do problema.

Figura 24 – Exemplo da falha buraco em diversos tamanhos de regiões de interesse.



(a) 101 px (b) 75 px (c) 51 px (d) 41 px (e) 31 px (f) 21 px (g) 11 px

Fonte: Autor.

Ao analisar a Figura 24, nota-se que a imagem à esquerda (a) possui grande quantidade de tons de cinza e à medida que o tamanho da região de interesse é reduzido, da esquerda para a direita, torna-se nítido que tal característica é perdida. Tal fato pode representar uma má identificação das falhas ao utilizar regiões de interesse muito pequenos.

3.7.3 Classificação das regiões de interesse

A classificação das regiões de interesse da imagem se divide em 4 etapas: localização das regiões de interesse na imagem, extração de atributos, classificação em falha ou não falha e segmentação das regiões que apresentam problemas na pele.

A primeira etapa, localização das regiões de interesse na imagem, é implementada após o pré-processamento e consiste na elaboração de uma lista com as localizações dos eixos X e Y das regiões de interesse na imagem original.

Após elaborada a lista de localizações, cada região de interesse da imagem original listado é analisado e, se no mesmo existir pelo menos um *pixel* na cor branca, a região de interesse é descartado (não analisado). Caso contrário, são extraídos os atributos a partir dos extratores GLCM ou LBP e gerado o seu vetor de características.

Tal procedimento descrito, é realizado porque a cor branca está situada nas regiões que envolvem o plano de fundo ou bordas da pele. A primeira região não faz parte da pele,

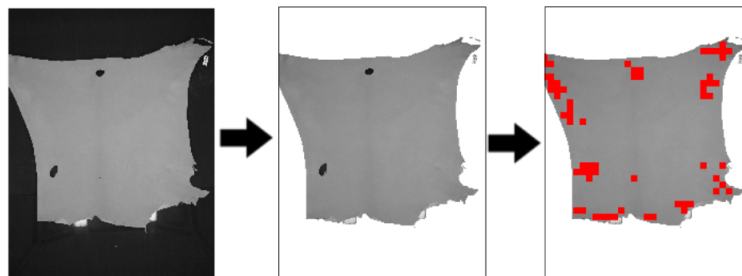
assim, não há a necessidade de fazer qualquer outro processamento nela. A última citada é descartada pelos curtumes, pois utilizam como matéria prima apenas as regiões centrais. Desse modo, também não se faz necessário classificar.

Nota-se que, ao descartar regiões que possuem *pixels* na cor branca, apenas a região de interesse será posteriormente classificada, ou seja, o que é realmente utilizado pelos curtumes.

Com os atributos extraídos, segue-se para a classificação das regiões de interesse a partir dos classificadores *Single Layer Perceptron*, *Logistic Regression*, *K-Nearest Neighbors* (KNN), Fisher (baseada na regra de Bayes), *Naïve Bayes*, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM). Após o processo de identificação do melhor classificador, este é utilizado no sistema que produz uma saída binária, na qual, se a região for considerada como isenta de defeito, será retornado o valor 0 e, em caso contrário, retornará o valor 1.

Após a aquisição do melhor classificador é iniciado o processo de segmentação das regiões de falha. O procedimento consiste em verificar os valores resultantes da classificação das regiões de interesse e, caso sejam detectados como defeito, receberão a coloração vermelha. De forma oposta, receberão a coloração dos *pixels* correspondentes da imagem original, tal processo é ilustrado na Figura 25.

Figura 25 – Exemplo da aplicação do algoritmo detector de falhas em regiões de interesse de 101 *pixels*.



Fonte: Autor.

Conforme pode ser verificado, esta funcionalidade é implementada com o intuito de destacar as falhas contidas em uma imagem de couro caprino. Portanto, neste estágio de evolução da presente Dissertação, já é possível considerar o auxílio aos especialistas na qualificação de couro caprino em uma de suas maiores dificuldades, a localização de regiões defeituosas na pele.

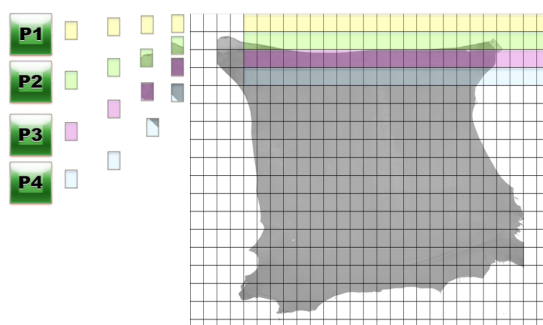
De acordo com Renato Pereira (2018), um especialista analisa uma peça de couro inteira em um tempo aproximado entre 10 a 30 segundos. Este período varia de acordo com o tamanho e a quantidade de falhas existentes na pele, bem como a capacidade e experiência do especialista.

Atualmente existem diversas técnicas que são voltadas para a redução do tempo

de processamento em algumas aplicações. O uso de *Graphics Processing Unit* (GPU) se mostra bastante promissor, porém tal técnica necessita de hardware específico. Deste modo, tendo em vista que a maioria dos computadores atuais possuem vários núcleos de processamento em sua arquitetura, nesta Dissertação faz-se o uso de computação paralela (explanada no Anexo C).

Como ilustrado na Figura 26, para a extração de atributos, classificação e segmentação das regiões de interesse, cada processo é responsável por uma linha ou conjunto de linhas da imagem carregada. Desta forma, a divisão ocorre de modo que, ao final, toda a peça de couro possa ser classificada e, caso haja necessidade, segmentada.

Figura 26 – Exemplo de imagem processada a partir do uso de 4 núcleos: P1, P2, P3 e P4.



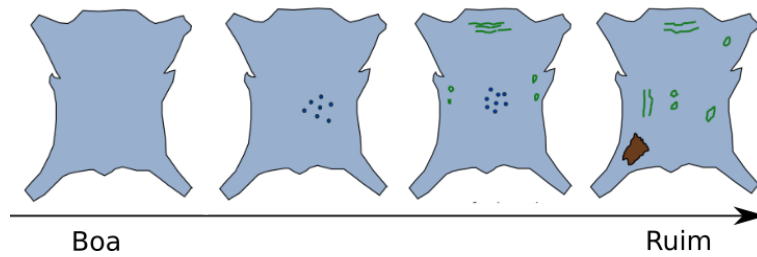
Fonte: Autor.

Com a aplicação do paralelismo, o conjunto horizontal de regiões de interesse (linhas) é direcionado para um determinado núcleo/processador. Desta forma, ao utilizar um computador *multicore* (com vários núcleos), o tempo de execução para a segmentação das regiões defeituosas é reduzido. Isto gera um melhor desempenho do programa e consequentemente, a redução no tempo de qualificação das peças de couro, fator de bastante importância para que sejam analisadas diversas peças em um curto período de tempo.

3.8 Localização das áreas livres de falhas

Conforme os especialistas em qualificação, é cada vez mais difícil obter peças de couro isentas de defeitos. Tendo em vista tal problema, conforme apresentado na Seção 2.2, a escolha das melhores peças de couro baseia-se no número, tamanho e localização de regiões sem defeitos (Figura 27).

Figura 27 – Exemplo de pele desejável (boa) e com diferentes níveis de defeitos.

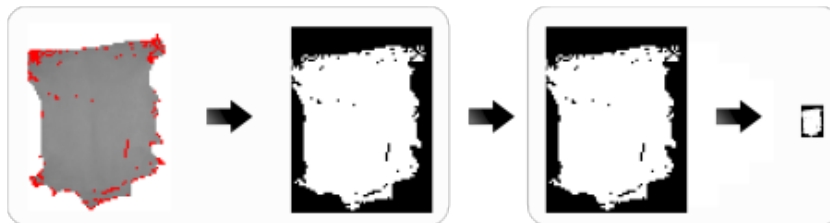


Fonte: Adaptado de Aslam et al. (2019).

Inicialmente, a imagem digital referente à amostra de couro caprino, com as regiões de falha já detectadas, é binarizada.

Após o procedimento descrito é elaborada uma matriz de referência. Esta é adotada levando em consideração cada região de interesse da imagem binarizada. Deste modo, caso a região de interesse possua *pixels* na cor branca, apenas um *pixel* referente a matriz elaborada também receberá a cor branca, em caso contrário, receberá a cor preta. O propósito de tal ação é possibilitar uma redução no tempo empregado para o processo de detecção de regiões isentas de falhas. Na Figura 28 é ilustrada tal ação.

Figura 28 – Binarização e elaboração da matriz de referência.



Fonte: Autor.

Com a matriz de representação elaborada, é dado início a detecção das regiões isentas de defeitos. Para isto são utilizados os melhores classificadores e tamanhos de regiões de interesse obtidos durante a detecção de falhas.

Salienta-se que, visando obter os melhores resultados relacionados a identificação da qualidade da peça de couro aliados ao menor tempo de execução, são utilizadas duas metodologias.

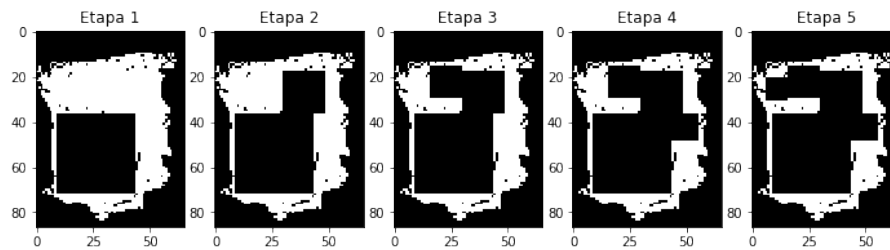
A primeira consiste em localizar diversas regiões livres de falhas com o critério de parada na análise de 80 % dos *pixels* isentos de defeitos da peça de couro. Tal valor é inserido pois a análise a partir de percentuais maiores mostrou-se desnecessária devido ao processamento de diversas regiões de interesse não são utilizadas pelos curtumes, como as bordas, por exemplo.

A segunda, utiliza como critério de parada a obtenção de 1, 5 e 10 regiões livres de falhas analisadas pelo algoritmo. Tais valores possuem o intuito da obtenção de resultados

semelhantes a metodologia anterior e em um tempo inferior.

Como forma de ilustrar a identificação das regiões livres de falha, na Figura 29 é apresentado o processo em 5 etapas. Salienta-se que essa implementação é descrita no Anexo D.

Figura 29 – Processo de detecção das regiões livres de falha.



Fonte: Autor.

A partir do processamento relacionado à localização das áreas livres de falhas com a combinação de classificadores e tamanhos de regiões de interesse, são formados diversos conjuntos de dados, sendo que cada um possuem 11 (onze) atributos. Estes são apresentados a seguir.

1. Área total sem falhas: tal atributo possui o intuito da obtenção da quantidade total de *pixels* isentos de falhas na peça de couro caprino;
2. Tamanho total da área processada sem defeitos: quantidade total de pixels identificados pelo algoritmo como regiões sem falhas. Nota-se que este valor deve ser o mais próximo possível do atributo 1. Neste sentido, o maior conjunto de *pixels* isentos de falhas serão detectados;
3. Tamanho da área isenta de defeitos não processada: a partir desse atributo é possível identificar a quantidade de *pixels* que não fazem parte das regiões detectadas pelo algoritmo. Esta característica é necessária quando se deseja localizar um número N de regiões isentas de falhas, pois, um valor alto pode representar uma grande quantidade de defeitos espalhados pela peça de couro caprino;
4. Regiões isentas de falhas detectadas: este atributo faz-se necessário quando deseja-se detectar regiões isentas de defeitos a partir de um percentual com base na área total sem falhas (atributo 1). Deste modo, quanto menor este valor, maiores serão as áreas detectadas;
5. Maior região detectada: para os curtumes quanto maior for uma região sem defeitos melhor será a peça de couro. Nesse sentido, esse atributo possui o intuito de retornar a maior região sem falhas.

6. Percentual da maior região detectada, em relação à área total sem falhas: este atributo é utilizado com o intuito de verificar a proximidade entre o tamanho total da peça de couro e a maior região detectada. Deste modo, esta característica possui o intuito de retornar o quão próximo em termos de tamanho é o atributo 5 do 1;
7. Menor área sem falhas detectadas: ao obter diversas regiões livres de falhas, se a menor região possuir um valor relativamente alto em comparação com a maior região detectada (atributo 5), pode-se considerar que o couro possui diversos defeitos espalhados pela peça.
8. Percentual da menor região detectada em relação à área total sem falhas: o seu uso objetiva obter um valor que possibilite analisar o tamanho da menor área detectada, em comparação com a quantidade total de *pixels* isentos de falha existentes na amostra de couro analisada.
9. Média aritmética referente a cada área livre de falhas detectada: ao analisar um número N de áreas livres de falhas e obter uma média aritmética muito baixa é possível identificar que a peça de couro analisada possui uma qualidade ruim, ou seja, contém diversos defeitos espalhados por toda a peça de couro.
10. Desvio padrão referente a cada área livre de falhas detectada: uma das características que definem um couro de qualidade é uma peça grande e isenta de defeitos. Deste modo, ao analisar diversas áreas isentas de defeitos e obter um valor baixo para este atributo pode-se denotar em uma pele de má qualidade.
11. Moda referente a cada área livre de falhas detectada: retorna a maior quantidade de tamanhos iguais de áreas isentas de falhas. Um valor alto pode representar que a peça de couro pode não fazer parte do nível de qualidade 1, ou seja, não possui uma área grande e isenta de defeitos.

Para o processo de aprendizado são utilizadas apenas as peças de couro que obtiveram o mesmo nível de qualidade pelos dois classificadores do curtume. Isto posto, o conjunto de imagens é dividido em 75 % do número de amostras da classe minoritária, ou seja, 6 peças de couro de cada classe para o treinamento e o restante do universo de imagens para os testes.

A partir dos diversos conjuntos de dados adquiridos, são iniciados os processos de treinamento fazendo uso dos classificadores *Logistic Regression*, *K-Nearest Neighbors* (KNN), Fisher, *Naïve Bayes*, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM), no qual apenas o melhor modelo computacional obtido é utilizado para identificar o nível de qualidade das peças de couro *wet blue*.

4 Resultados e discussões

Para a qualificação das peças de couro, há a necessidade de três procedimentos: o pré-processamento, a detecção de falhas e, por fim, a definição do nível de qualidade. À vista disto, neste Capítulo são apresentados os resultados obtidos relacionados aos itens citados.

4.1 Resultados do pré-processamento das amostras de couro caprino

Uma problemática a ser resolvida durante o pré-processamento, além da remoção do plano de fundo, está ligada ao tipo de falha relacionado aos buracos nas peças de couro. Estes defeitos possuem a mesma tonalidade do plano de fundo, porém devem ser distinguidos pelo sistema, pois, devem ser detectados como regiões de falha.

Visando remover o plano de fundo e resolver o problema em questão, como citado na Seção 3.7, são utilizadas três técnicas de pré-processamento. Na Figura 30, como exemplo, é apresentado uma amostra de resultado de cada técnica utilizada para a mesma imagem.

Figura 30 – Técnicas aplicadas para remoção do plano de fundo.



Fonte: Autor.

A primeira técnica, baseada em crescimento de regiões, obtêm um tempo médio de aproximadamente 2,67 segundos e consegue remover o plano de fundo e manter o objeto de interesse em todas as amostras do conjunto de imagens. A partir da Figura 30 (a), pode-se perceber que o uso dessa técnica possibilita a remoção de todo o plano de fundo diferenciando-o do tipo de falha relacionada aos buracos.

Ao aplicar a segunda metodologia, baseada nos processos de erosão e dilatação, a partir da combinação das duas técnicas, é obtido um tempo médio de 1,48 segundos. No

entanto, ao aplicá-la em amostras de couro que possuem o tipo de falha relacionado a buracos, faz-se necessário diversas combinações de número de execuções destes processos. Dessa forma, a quantidade de execuções dos métodos de erosão ou dilatação dependem do tamanho da falha em questão contida na peça de couro. Assim, para buracos muito grandes, o resultado pode variar entre o surgimento de regiões com tonalidades escuras nas bordas da peça de couro caprino (Figura 30 (b)) ou exclusões de partes importantes da imagem referente a peça de couro o que pode gerar uma qualificação equivocada das peças de couro.

A terceira metodologia, baseada na detecção de contornos (Figura 30 (c)), apresenta um tempo médio de 1,58 segundos e extrai o plano de fundo, mantendo apenas o objeto de interesse em todas as imagens utilizadas. Dessa forma, por diferenciar bem a peça de couro do plano de fundo, não apresentar ruídos nas imagens e ser a técnica com o segundo melhor resultado relacionado ao tempo de execução, é a metodologia escolhida para o processamento das demais etapas nas quais os resultados são apresentados a seguir.

4.2 Resultados obtidos na classificação binária

Nesta Seção são apresentados os resultados obtidos da classificação binária (normal ou falha) dos conjuntos de dados. Estes são elaborados a partir de um arquivo com as localizações em amostras de couro caprino, citados na Seção 3.3.

Para extração dos atributos são adotadas regiões de interesse com dimensões de 11 x 11, 21 x 21, 31 x 31, 41 x 41, 51 x 51, 75 x 75 e 101 x 101 *pixels* e os extratores *Local Binary Patterns* (LBP) e *Gray Level Co-Occurrence Matrix* (GLCM).

Salienta-se que, conforme apresentado na Seção 3.4, são utilizadas 5 configurações de parâmetros para o extrator LBP, logo, apenas a melhor configuração é apresentada de acordo com cada classificador.

Quanto a escolha dos melhores resultados, estes baseiam-se na combinação entre maior taxa de acerto e equilíbrio entre o conjunto de treinamento e teste.

4.2.1 Resultados médios obtidos a partir do treinamento entre amostras normais e amostras de falha

Com o intuito de obter a melhor combinação entre dimensões de regiões de interesse, extratores e classificadores, diversos treinamentos são executados exaustivamente, a partir dos parâmetros citados na Seção 3.5, e os melhores resultados médios obtidos a partir de 50 execuções são descritos a seguir.

Perceptron de Única Camada

Como citado na Seção 3.5, o Perceptron é aplicado nesta Dissertação com o intuito de verificar se é possível realizar separação entre as classes com uma superfície de decisão linear.

Com a aplicação desse classificador, pode-se obter, a partir do extrator de características LBP, uma acurácia média de 91,37 % no treinamento e 91,08 % no teste. Na Tabela 11, para fins comparativos, são apresentados os resultados médios obtidos com uso dos extratores LBP e GLCM.

Tabela 11 – Melhores resultados médios obtidos a partir do perceptron simples.

região de interesse (px)	$LBP_{1,8}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	87,21 $\pm$ 3,65	87,49 $\pm$ 2,70	89,58 $\pm$ 0,54	56,94 $\pm$ 11,28
21 x 21	91,37 $\pm$ 2,55	91,08 $\pm$ 1,59	91,08 $\pm$ 0,48	77,48 $\pm$ 2,08
31 x 31	88,19 $\pm$ 3,12	88,64 $\pm$ 2,27	88,50 $\pm$ 0,56	78,32 $\pm$ 1,50
41 x 41	85,62 $\pm$ 3,78	85,85 $\pm$ 2,18	84,88 $\pm$ 0,72	76,96 $\pm$ 1,22
51 x 51	82,30 $\pm$ 5,75	81,94 $\pm$ 2,76	82,01 $\pm$ 0,61	77,46 $\pm$ 0,69
75 x 75	81,46 $\pm$ 4,61	80,56 $\pm$ 2,64	77,46 $\pm$ 0,69	71,46 $\pm$ 1,69
101 x 101	79,30 $\pm$ 6,05	78,72 $\pm$ 2,91	75,01 $\pm$ 0,84	69,19 $\pm$ 1,50

Salienta-se que para este classificador, a melhor configuração para o extrator LBP foi obtida a partir do número de pontos igual a 8 e raio igual a 1. Deste modo, a melhor acurácia considerada está destacada em negrito.

Logistic Regression

Quanto ao processo de classificação com *Logistic Regression*, percebeu-se que em ambos os extratores utilizados, LBP e GLCM, a melhor dimensão de região de interesse é de 21 x 21 *pixels*. Como melhor resultado médio, é obtida uma acurácia de 97,17 % no treinamento e 97,03 % no teste, a partir do extrator GLCM. Estes resultados são apresentados na Tabela 12.

Tabela 12 – Melhores resultados médios obtidos a partir do *Logistic Regression*.

região de interesse (px)	$LBP_{2,16}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	89,16 $\pm$ 0,31	88,98 $\pm$ 0,69	93,98 $\pm$ 0,30	92,67 $\pm$ 0,70
21 x 21	90,95 $\pm$ 0,38	90,11 $\pm$ 0,74	97,15 $\pm$ 0,15	97,03 $\pm$ 0,35
31 x 31	89,85 $\pm$ 0,34	88,55 $\pm$ 0,78	96,73 $\pm$ 0,22	96,13 $\pm$ 0,44
41 x 41	88,54 $\pm$ 0,34	86,81 $\pm$ 0,73	95,77 $\pm$ 0,21	95,09 $\pm$ 0,57
51 x 51	87,52 $\pm$ 0,46	85,45 $\pm$ 0,95	94,75 $\pm$ 0,32	93,63 $\pm$ 0,56
75 x 75	86,35 $\pm$ 0,38	83,84 $\pm$ 0,78	92,52 $\pm$ 0,34	91,19 $\pm$ 0,77
101 x 101	85,56 $\pm$ 0,43	82,69 $\pm$ 0,94	90,99 $\pm$ 0,34	89,20 $\pm$ 0,88

Salienta-se que em relação do extrator LBP, os melhores resultados foram obtidos a partir dos parâmetros de raio igual a 2 e número de pontos igual a 16, sendo estes, inferiores aos resultados obtidos com a utilização do extrator GLCM.

K-Nearest Neighbors (KNN)

No processo de classificação realizado com o classificador K-NN, o melhor resultado é obtido a partir dados provindos do extrator GLCM, parâmetro K igual a 5, obtendo uma acurácia média de 97,71 % no treinamento e 96,31 % no teste. Na Tabela 13 são apresentados os melhores resultados médios obtidos com a aplicação dos extratores LBP e GLCM a partir da utilização deste classificador.

Tabela 13 – Melhores resultados médios obtidos a partir do KNN.

região de interesse (px)	$LBP_{1.5,12}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	90,47 $\pm$ 0,43	86,66 $\pm$ 0,87	94,55 $\pm$ 0,29	92,39 $\pm$ 0,62
21 x 21	91,95 $\pm$ 0,36	87,87 $\pm$ 0,75	97,71 $\pm$ 0,23	96,31 $\pm$ 0,46
31 x 31	90,31 $\pm$ 0,38	85,33 $\pm$ 0,78	97,39 $\pm$ 0,24	94,31 $\pm$ 0,58
41 x 41	88,95 $\pm$ 0,45	82,37 $\pm$ 1,01	96,50 $\pm$ 0,28	91,81 $\pm$ 0,68
51 x 51	87,23 $\pm$ 0,51	80,53 $\pm$ 0,99	95,60 $\pm$ 0,28	89,96 $\pm$ 0,89
75 x 75	85,44 $\pm$ 0,45	78,34 $\pm$ 0,85	89,82 $\pm$ 0,39	84,53 $\pm$ 0,84
101 x 101	83,77 $\pm$ 0,46	76,80 $\pm$ 0,89	88,07 $\pm$ 0,37	82,11 $\pm$ 0,88

Observa-se que, para ambos os extratores o uso da dimensão da região de interesse referente 21 x 21 *pixels* proporciona os melhores resultados.

Classificador de Fisher

Quanto a utilização do classificador de Fisher, o melhor resultado médio é obtido a partir da utilização de dados provindos do extrator GLCM e regiões de interesse com dimensões de 21 x 21 *pixels*. Na Tabela 14 são apresentados os resultados obtidos por meio do uso desse classificador.

Tabela 14 – Melhores resultados médios obtidos a partir do classificador de Fisher.

região de interesse (px)	$LBP_{1,5,12}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	88,92 $\pm$ 0,32	87,70 $\pm$ 0,72	92,18 $\pm$ 0,30	88,87 $\pm$ 0,70
21 x 21	91,98 $\pm$ 0,32	90,34 $\pm$ 0,74	96,38 $\pm$ 0,20	94,86 $\pm$ 0,45
31 x 31	89,64 $\pm$ 0,43	86,59 $\pm$ 0,87	96,44 $\pm$ 0,26	95,44 $\pm$ 0,45
41 x 41	88,34 $\pm$ 0,33	84,61 $\pm$ 0,72	96,11 $\pm$ 0,25	94,56 $\pm$ 0,43
51 x 51	86,30 $\pm$ 0,46	81,46 $\pm$ 0,97	95,19 $\pm$ 0,23	93,62 $\pm$ 0,42
75 x 75	84,87 $\pm$ 0,48	79,48 $\pm$ 1,06	93,72 $\pm$ 0,39	90,95 $\pm$ 0,60
101 x 101	82,95 $\pm$ 0,49	76,54 $\pm$ 0,88	91,50 $\pm$ 0,34	88,02 $\pm$ 0,73

Conforme apresentado em destaque, nota-se que a melhor classificação média é obtida a partir dos dados advindos do extrator GLCM e dimensões de regiões de interesse iguais a 31 x 31 *pixels*, obtendo uma acurácia de 96,44 % no treinamento e 95,44 % no teste.

Naïve Bayes

Para o processo realizado a partir do classificador *Naïve Bayes*, enfatiza-se que durante o uso do extrator LBP, a configuração que apresentou melhores resultados é obtida a partir dos parâmetros de raio igual a 1,5 e número de pontos igual a 12. No entanto, os melhores resultados são obtidos a partir da utilização do extrator GLCM obtendo 96,70 % no treinamento e 96,55 % no teste. Na Tabela 15 são apresentados os demais valores obtidos a partir do uso deste classificador.

Tabela 15 – Melhores resultados médios obtidos a partir do *Naïve Bayes*.

região de interesse (px)	$LBP_{1.5,12}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	88,08 $\pm$ 0,36	87,71 $\pm$ 0,83	93,23 $\pm$ 0,32	90,65 $\pm$ 0,73
21 x 21	90,89 $\pm$ 0,35	88,78 $\pm$ 0,82	96,70 $\pm$ 0,22	96,55 $\pm$ 0,43
31 x 31	88,51 $\pm$ 0,37	85,41 $\pm$ 0,92	96,22 $\pm$ 0,26	95,31 $\pm$ 0,42
41 x 41	87,18 $\pm$ 0,31	83,54 $\pm$ 0,68	92,64 $\pm$ 0,41	90,10 $\pm$ 0,73
51 x 51	86,10 $\pm$ 0,38	81,76 $\pm$ 0,78	89,40 $\pm$ 0,41	85,23 $\pm$ 0,81
75 x 75	82,83 $\pm$ 0,44	78,33 $\pm$ 1,06	84,46 $\pm$ 0,37	78,10 $\pm$ 0,94
101 x 101	80,84 $\pm$ 0,37	75,48 $\pm$ 0,83	80,84 $\pm$ 0,42	72,80 $\pm$ 0,82

Com base no exposto, torna-se claro que os melhores resultados foram alcançados a partir dos conjuntos advindos do extrator GLCM e dimensões de regiões de interesse iguais a 21 x 21 *pixels*.

MultiLayer Perceptron (MLP)

Semelhante aos demais resultados obtidos a partir das utilizações de classificadores não lineares, o melhor resultado com a utilização do classificador MLP é obtido com dados adquiridos do extrator GLCM, alcançando uma acurácia de 98,31 % no treinamento e 97,75 % no teste. Os demais resultados são apresentados na Tabela 16.

Tabela 16 – Melhores resultados médios obtidos a partir do MLP.

região de interesse (px)	$LBP_{1,5,12}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	94,45 $\pm$ 0,56	82,72 $\pm$ 1,18	96,62 $\pm$ 1,70	94,55 $\pm$ 3,22
21 x 21	99,63 $\pm$ 0,15	88,18 $\pm$ 0,80	98,16 $\pm$ 0,31	96,69 $\pm$ 0,16
31 x 31	98,48 $\pm$ 0,50	86,60 $\pm$ 0,79	98,31 $\pm$ 0,42	97,75 $\pm$ 0,33
41 x 41	94,79 $\pm$ 0,67	85,12 $\pm$ 0,97	97,86 $\pm$ 0,61	96,63 $\pm$ 0,40
51 x 51	92,97 $\pm$ 0,59	82,95 $\pm$ 1,21	94,95 $\pm$ 1,03	95,05 $\pm$ 1,67
75 x 75	99,11 $\pm$ 1,45	80,63 $\pm$ 1,07	97,58 $\pm$ 0,91	95,69 $\pm$ 2,37
101 x 101	91,71 $\pm$ 0,81	80,50 $\pm$ 1,15	95,56 $\pm$ 1,71	95,13 $\pm$ 2,51

Salienta-se que o resultado em destaque refere-se a utilização de regiões de interesse de tamanho iguais a 31 x 31 *pixels*.

A topologia para a obtenção dos melhores resultados, são apresentadas a seguir.

- Taxa de aprendizado inicial da camada oculta: 0,3;
- Sem decaimento da taxa de aprendizado da camada oculta;
- Taxa de aprendizado da camada de saída: 0,3;
- Decaimento linear da taxa de aprendizado da camada de saída;
- Quantidade de neurônios na camada oculta: 21;
- Faixa de rejeição: 0,1;
- Função de ativação: logística.

Support Vector Machine (SVM)

Por fim, a cerca do classificador SVM, os melhores resultados são obtidos a partir da adoção de dados providos de dimensões de regiões de interesse de 21 x 21 *pixels* e extrator GLCM. Com esta configuração pode-se obter uma acurácia de 97,19 % no treinamento e 97,20 % no teste.

A seguir, na Tabela 17, são apresentados os melhores resultados médios obtidos a partir da utilização desse classificador.

Tabela 17 – Melhores resultados médios obtidos a partir do SVM.

região de interesse (px)	$LBP_{1,5,12}$		GLCM	
	Acurácia Média (%) / Desvio Padrão (σ)			
	Treinamento	Teste	Treinamento	Teste
11 x 11	87,05 $\pm$ 0,57	88,23 $\pm$ 0,74	91,27 $\pm$ 0,35	92,81 $\pm$ 0,61
21 x 21	94,93 $\pm$ 0,27	91,83 $\pm$ 0,72	97,19 $\pm$ 0,18	97,20 $\pm$ 0,40
31 x 31	90,30 $\pm$ 0,36	88,17 $\pm$ 0,90	97,25 $\pm$ 0,20	96,84 $\pm$ 0,41
41 x 41	89,16 $\pm$ 0,31	86,63 $\pm$ 0,83	96,31 $\pm$ 0,25	96,02 $\pm$ 0,51
51 x 51	86,11 $\pm$ 0,62	83,27 $\pm$ 0,81	96,59 $\pm$ 0,17	95,95 $\pm$ 0,38
75 x 75	85,34 $\pm$ 0,71	82,54 $\pm$ 1,84	94,85 $\pm$ 0,30	93,61 $\pm$ 0,72
101 x 101	84,73 $\pm$ 0,44	83,95 $\pm$ 0,77	92,44 $\pm$ 0,39	90,45 $\pm$ 0,81

Salienta-se que para o classificador citado, é utilizado o *kernel* linear, 326 vetores de suporte e parâmetro C igual a 10.

4.2.2 Análise dos melhores resultados obtidos pelos classificadores

Como apresentado, os melhores resultados médios são obtidos a partir dos classificadores *Logistic Regression*, *Multilayer Perceptron* e *Support Vector Machine*. Neste sentido, após 50 execuções de cada melhor classificador encontrado, são obtidos os três melhores resultados gerais. Estes são apresentados nas Tabelas 18, 19 e 20, a partir das suas matrizes de confusão.

Tabela 18 – Matriz de confusão proveniente do processo de treinamento e teste com *Logistic Regression*, dados obtidos a partir do extrator GLCM e regiões de interesse de 21 x 21 *pixels*.

Real \ Predito	Treinamento		Teste	
	Normal	Falha	Normal	Falha
Normal	48,80 %	1,20 %	26,18 %	0,56 %
Falha	1,76 %	48,24 %	1,76 %	71,50 %

Tabela 19 – Matriz de confusão proveniente do processo de treinamento e teste com MLP, dados obtidos a partir do extrator GLCM e regiões de interesse de 31 x 31 *pixels*.

Real \ Predito	Treinamento		Teste	
	Normal	Falha	Normal	Falha
Normal	49,44 %	0,56 %	19,53 %	0,48 %
Falha	0,86 %	49,14 %	1,38 %	78,61 %

Tabela 20 – Matriz de confusão proveniente do processo de treinamento e teste com SVM, dados obtidos a partir do extrator GLCM e regiões de interesse de 21 x 21 *pixels*.

Real \ Predito	Treinamento		Teste	
	Normal	Falha	Normal	Falha
Normal	48,80 %	1,20 %	26,58 %	0,16 %
Falha	1,76 %	48,24 %	2,08 %	71,78 %

Como ilustrado na Seção 3.5, para o processo de teste são utilizados conjuntos de dados desbalanceados. Desse modo, na Tabela 21, com o intuito de obter uma melhor análise a cerca da classificação em cada tipo de classe, são apresentados os resultados referentes a precisão, Recall e *F1*.

Tabela 21 – Valores referentes à precisão, *recall* e F1 relacionados aos melhores modelos obtidos.

Clas.	Ext.	Dim. Região	Etapa	Acurácia	Classe	Precisão	<i>Recall</i>	<i>F1</i>
LR	GLCM	21 x 21 <i>pixels</i>	Treinamento	0,9704	Normal Falha	0,9652 0,9758	0,9760 0,9648	0,9706 0,9702
			Teste	0,9768	Normal Falha	0,9370 0,9922	0,9790 0,9760	0,9575 0,9840
MLP	GLCM	31 x 31 <i>pixels</i>	Treinamento	0,9858	Normal Falha	0,9829 0,9887	0,9888 0,9828	0,9858 0,9857
			Teste	0,9814	Normal Falha	0,9341 0,9939	0,9760 0,9828	0,9546 0,9883
SVM	GLCM	21 x 21 <i>pixels</i>	Treinamento	0,9704	Normal Falha	0,9652 0,9758	0,9760 0,9648	0,9706 0,9702
			Teste	0,9776	Normal Falha	0,9274 0,9978	0,9940 0,9716	0,9595 0,9845

Ao analisar os resultados apresentados é possível perceber que o classificador MLP possui os melhores resultados de classificação e equilíbrio entre amostras normais e amostras de falha. No entanto, os classificadores LR e SVM obtiveram resultados próximos do melhor classificador citado. Deste modo, devido a esta proximidade, ambos os classificadores são utilizados nas Seções seguintes.

Destaca-se ainda que a classificação dos dados obtidos a partir do extrator LBP foram interiores aos obtidos fazendo uso do GLCM. Deste modo, estes não são apresentados nas próximas etapas deste trabalho.

4.3 Análise dos erros de classificação por tipo de falha

Visando ilustrar os melhores processos de classificação obtidos em relação a cada tipo de falha ou região normal, na Tabela 22 são apresentadas as quantidades de erros de classificação a partir de cada agrupamento elaborado (apresentados na Tabela 6).

Tabela 22 – Análise da distribuição dos erros de classificação durante o treinamento e teste.

Ag	Am	<i>LR</i>			Perc.	<i>MLP</i>			Perc.	<i>SVM</i>			Perc.
		Trein	Test	Tot		Trein	Test	Tot		Trein	Test	Tot	
0	1699	34	6	40	2.40 %	10	10	20	1.20 %	34	3	37	2.22 %
1	477	0	2	2	0.42 %	0	2	2	0.42 %	1	2	3	0.63 %
2	97	0	0	0	0.00 %	0	0	0	0.00 %	0	0	0	0.00 %
3	93	3	1	4	4.30 %	2	0	2	2.15 %	4	0	4	4.30 %
4	462	24	12	36	7.79 %	18	12	30	6.49 %	27	11	38	8.23 %
5	168	1	0	1	0.60 %	1	0	1	0.60 %	0	1	1	0.60 %
6	253	2	2	4	1.58 %	0	0	0	0.00 %	1	2	3	1.19 %
7	20	0	0	0	0.00 %	0	0	0	0.00 %	0	0	0	0.00 %
8	46	2	0	2	4.35 %	1	3	4	8.70 %	1	1	2	4.35 %
9	252	4	3	7	2.78 %	4	2	6	2.38 %	3	3	6	2.38 %
10	317	7	2	9	2.83 %	1	1	2	0.63 %	7	3	10	3.15 %
11	65	2	1	3	4.62 %	1	1	2	3.08 %	1	2	3	4.62 %

Como ilustrado, é nítido que a maior problemática surge a partir do tipo de falha 4. Dessa forma, a falha referente ao corte leve de faca apresenta mais dificuldades para uma correta identificação. De maneira oposta, a classificação relacionada aos grupos 2 (sinais mais graves) e 7 (buracos e bexigas) apresentaram ótimos resultados.

Outra dificuldade é evidenciada com o classificador MLP no agrupamento 8, ou seja, na falha relacionada a cicatrizes. Nesta, nota-se um resultado inferior ao valor obtido na classificação de falhas relacionadas a cortes leve de faca.

Conclui-se nesta análise que, apesar da existência de uma grande quantidade de amostras, evidencia-se uma maior dificuldade em identificar falhas relacionadas ao corte leve de faca, visto a quantidade de erros de classificação em comparação as demais falhas detectadas pelo sistema.

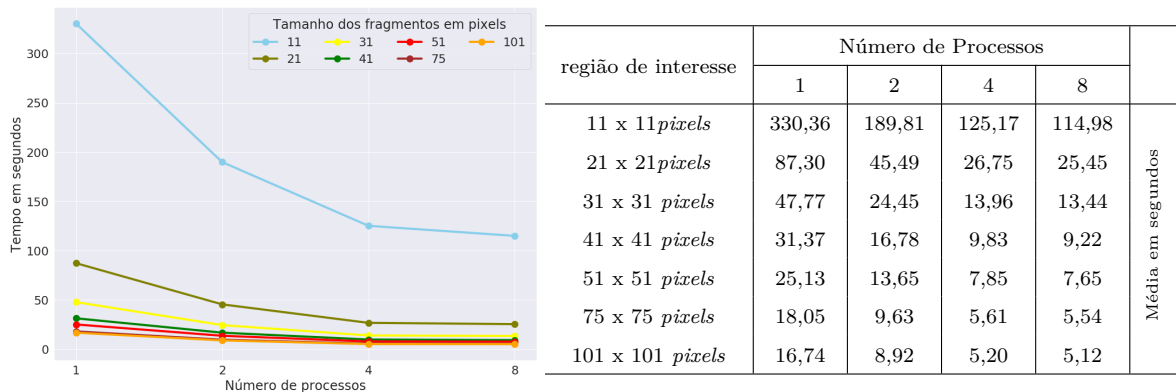
4.4 Tempo de execução da segmentação de acordo com classificador e dimensão da região de interesse

Conforme explanado na Seção 4.2, os melhores resultados obtidos são adquiridos, na maioria dos casos, a partir de regiões de interesse com dimensões iguais a 21 x 21 ou 31 x 31 *pixels*. Todavia, a utilização de regiões de interesse maiores apresentam um menor tempo de execução. Dessa forma, a seguir são apresentados o tempo médio de pré-processamento e detecção de falhas em couro caprino com tamanhos de regiões de interesse iguais a 11 x 11, 21 x 21, 31 x 31, 41 x 41, 51 x 51, 75 x 75 e 101 x 101 *pixels*.

Salienta-se que as execuções de pré-processamento e detecção de falhas são executadas em um computador com processador Intel Core i5-7300HQ CPU 2.50 GHz x 4, 8 GB RAM DDR 4 2666 MHz (8 GB x 1), HD de 1 TB 5400 RPM.

Quanto ao processo de execução, o mesmo ocorre a partir do uso do sistema de forma não paralelizada, ou seja, com apenas 1 processo sendo utilizado em toda a detecção de falhas e, com o intuito de executar em um menor tempo, utilizado de forma paralelizada, com 2, 4 e 8 processos por execução.

Na Tabela 23 é apresentado o tempo médio obtido durante a detecção de falhas a partir da classificação com uso do algoritmo *Logistic Regression* em 50 execuções.

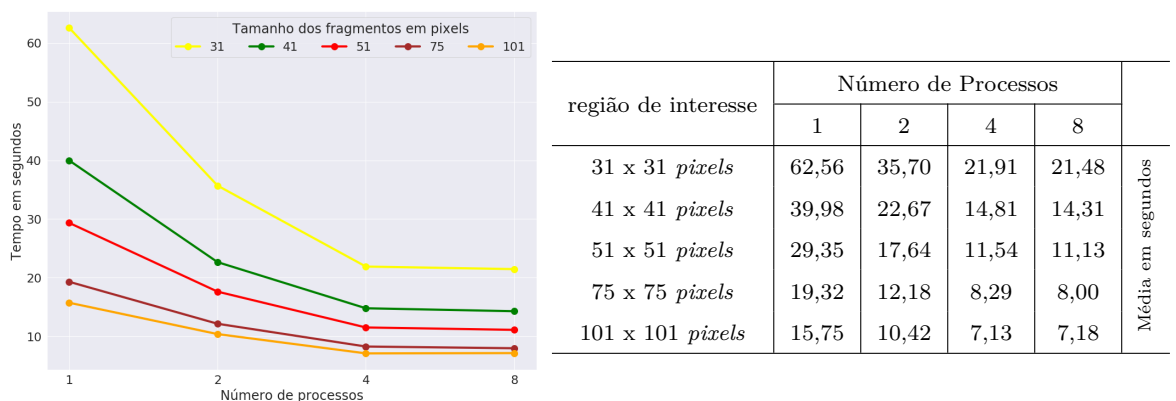
Tabela 23 – Tempo médio para detecção de falhas em couro - *Logistic Regression*.

Percebe-se que quanto menor a dimensão da região de interesse, mais tempo será necessário para analisar uma imagem digital. A explicação para que isto ocorra baseia-se no fato de que, quanto menor o tamanho das regiões de interesse, mais deles deverão existir para compor a imagem inteira. Deste modo, mais regiões serão analisadas.

Como apresentado na Seção 3.7, um profissional da área de qualificação de peças de couro leva em média de 10 a 30 segundos para identificar os defeitos e, posteriormente, qualificar uma pele inteira. Neste sentido, a partir desta Seção serão apresentados apenas as dimensões de regiões de interesse superiores a 21 x 21 pixels, pois, a partir do uso deles é obtido um tempo aproximado com a análise visual executada nos curtumes.

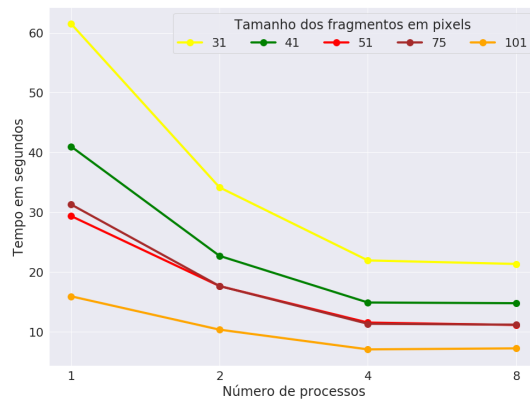
Na Tabela 24, são apresentados os tempos médio obtidos da implementação com o classificador MLP.

Tabela 24 – Tempo médio para detecção de falhas em couro - MLP.



Na Tabela 25 são apresentados os tempos médio obtidos da implementação com o classificador SVM.

Tabela 25 – Tempo médio para detecção de falhas em couro - SVM.



região de interesse	Número de Processos				Média em segundos
	1	2	4	8	
31 x 31 pixels	61,51	34,15	21,93	21,34	
41 x 41 pixels	40,98	22,70	14,90	14,78	
51 x 51 pixels	31,29	17,65	11,33	11,19	
75 x 75 pixels	22,92	11,80	8,22	8,08	
101 x 101 pixels	15,93	10,35	7,05	7,23	

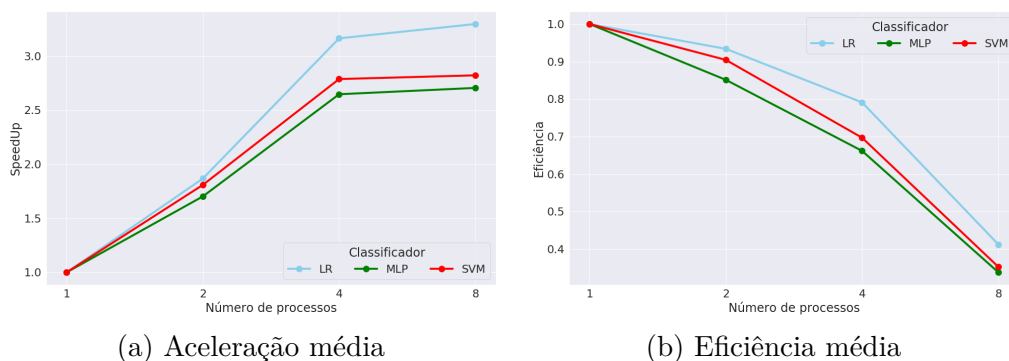
Nota-se que em relação ao tempo, não ocorrem diferenças relevantes quanto ao uso de determinado classificador. Dessa forma, pode-se afirmar que todos possuem o mesmo desempenho.

Quanto ao uso das regiões de interesse, é evidenciado que quanto menor o tamanho deles, maior é o tempo para a detecção das falhas. Em contrapartida, a utilização das menores regiões de interesse escolhidas, ou seja, 31 x 31 pixels, possuem um tempo similar ao utilizado nos curtumes e apresentam os melhores resultados de classificação dentre os tamanhos utilizados, sendo assim, estes também serão analisados nas etapas seguintes.

4.4.1 Análise da aceleração média e eficiência na aplicação do paralelismo

Na Figura 31 é apresentada a aceleração média e a eficiência referente à aplicação do paralelismo. Na Subfigura (a) é possível perceber a obtenção do desempenho relacionado entre 2 e 3 processadores a partir da utilização de 4 núcleos. Nota-se que ao aumentar o número de processos, este resultado tende a manter-se.

Figura 31 – Resultado médio da aplicação do paralelismo.



Fonte: Autor.

Analisando a eficiência média (Figura 31 (b)), torna-se nítido que à medida que o número de processos aumenta, apesar de reduzir a quantidade de tempo necessário para a resolução do problema, reduz-se a eficiência do sistema.

Salienta-se que se o número de processos ultrapassar a quantidade de processadores contidos na máquina, o tempo tende a estagnar-se. A justificativa para que isto ocorra, parte do princípio de que quanto mais processos são utilizados, maior a fila para gerenciamento dos mesmos. Isto deve-se ao fato de ocorrer uma sobrecarga de comunicação, pois, devido à grande quantidade de processos, há uma perda de tempo para o sincronismo entre as partes.

Ao decompor bastante os dados a serem analisados, ocorre um problema de concorrência entre núcleos de processamento. Dessa forma, a latência de transferência pode não ser o suficiente para o trabalho exigido.

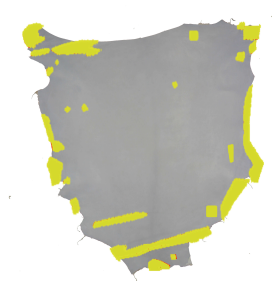
Ao analisar os resultados provindos da aceleração e da eficiência, percebe-se que os melhores resultados são obtidos a partir da utilização do classificador *Logistic Regression*, isto deve-se a simplicidade do algoritmo, o que proporciona um menor tempo para a detecção de falhas.

4.5 Análise da detecção e classificação de regiões de interesse em amostras de couro caprino

Um dos métodos para analisar se as soluções propostas de fato generalizam bem o problema é testá-las de forma prática. Dessa forma, representações de imagens contendo peças de couro são adquiridas e processadas por cada topologia escolhida.

Para fins de comparação sobre o nível de qualidade das áreas segmentadas através dos algoritmos propostos, algumas imagens são analisadas a olho nu e, ao verificar falhas na peça de couro, com uso do *software* editor de imagens Gimp, essas regiões são coloridas na cor amarela. Na Figura 32 é apresentado um exemplo dessas análises.

Figura 32 – Destaque de falhas a partir de uma análise a olho nu.

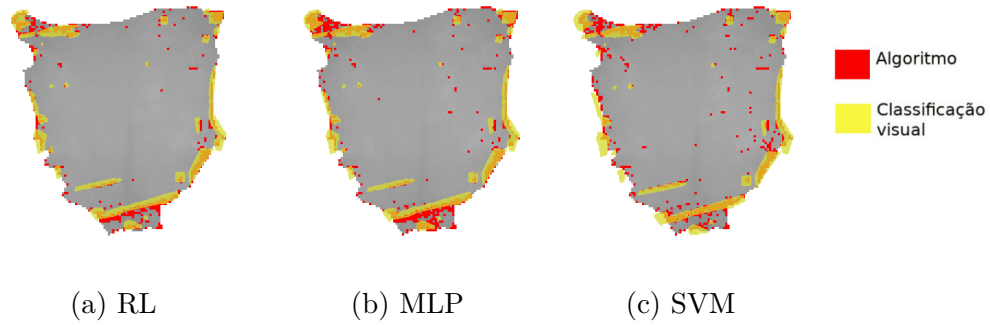


Fonte: Autor.

Nas Figuras 33, 34, 35, 36 e 37 é possível verificar o resultado do processamento através das imagens obtidas em diversas dimensões de regiões de interesse, a partir dos algoritmos propostos, e compará-las com a imagem analisada a olho nu.

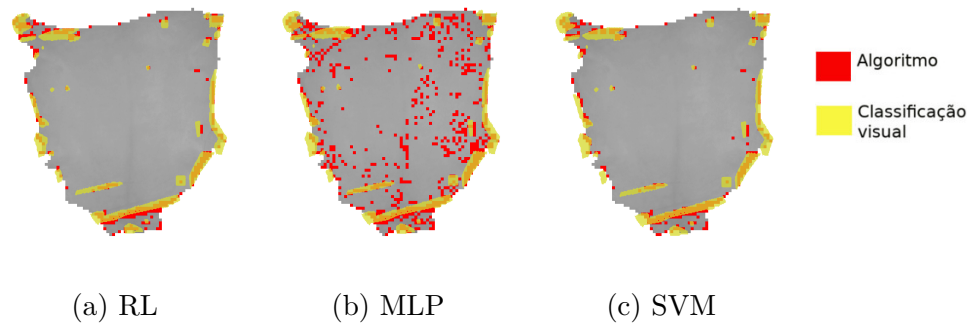
As regiões de interesse destacadas na cor amarela representam o resultado da detecção visual e, na cor vermelha, a partir dos algoritmos utilizados.

Figura 33 – Exemplo de detecção de falhas com regiões de interesse de 31 x 31 *pixels*.



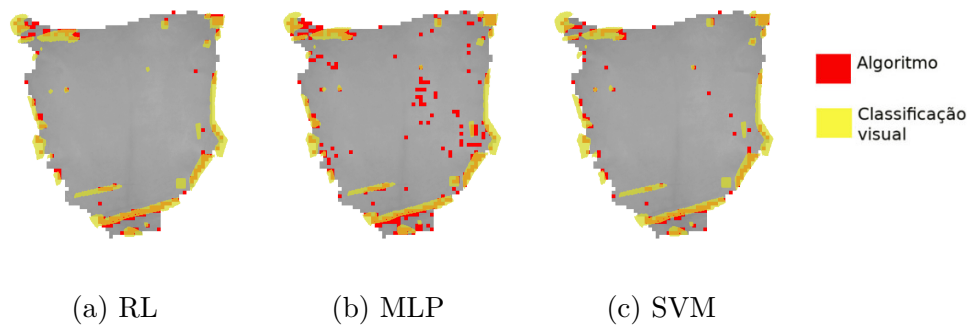
Fonte: Autor.

Figura 34 – Exemplo de detecção de falhas com regiões de interesse de 41 x 41 *pixels*.



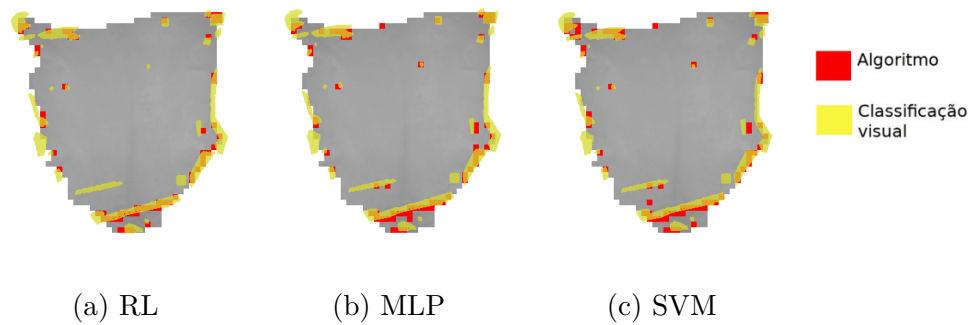
Fonte: Autor.

Figura 35 – Exemplo de detecção de falhas com regiões de interesse de 51 x 51 *pixels*.



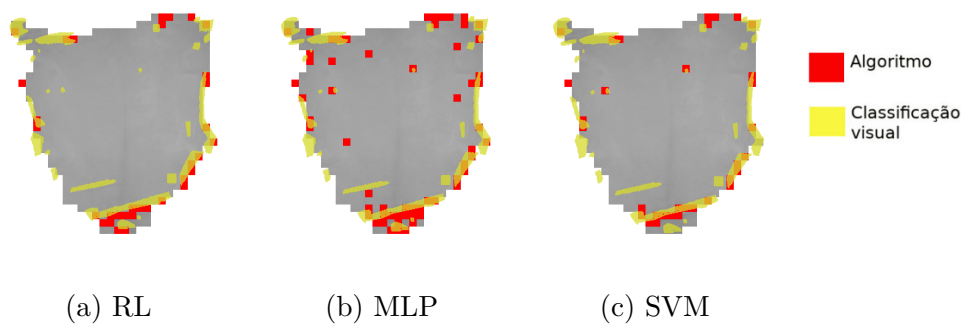
Fonte: Autor.

Figura 36 – Exemplo de detecção de falhas com regiões de interesse de 75 x 75 *pixels*.



Fonte: Autor.

Figura 37 – Exemplo de detecção de falhas com regiões de interesse de 101 x 101 *pixels*.



Fonte: Autor.

A utilização de regiões de interesse com dimensões iguais a 31 x 31 *pixels* resultou em diversos ruídos durante a detecção de falhas. Percebe-se ainda que o uso do classificador MLP se mostrou de forma bastante diferente das análises realizadas pelo especialista do curteme. Em contrapartida, as detecções realizadas a partir do uso dos classificadores *Logistic Regression* e SVM possuem resultados similares à análise visual.

4.6 Resultados obtidos durante o processo de qualificação das peças de couro caprino

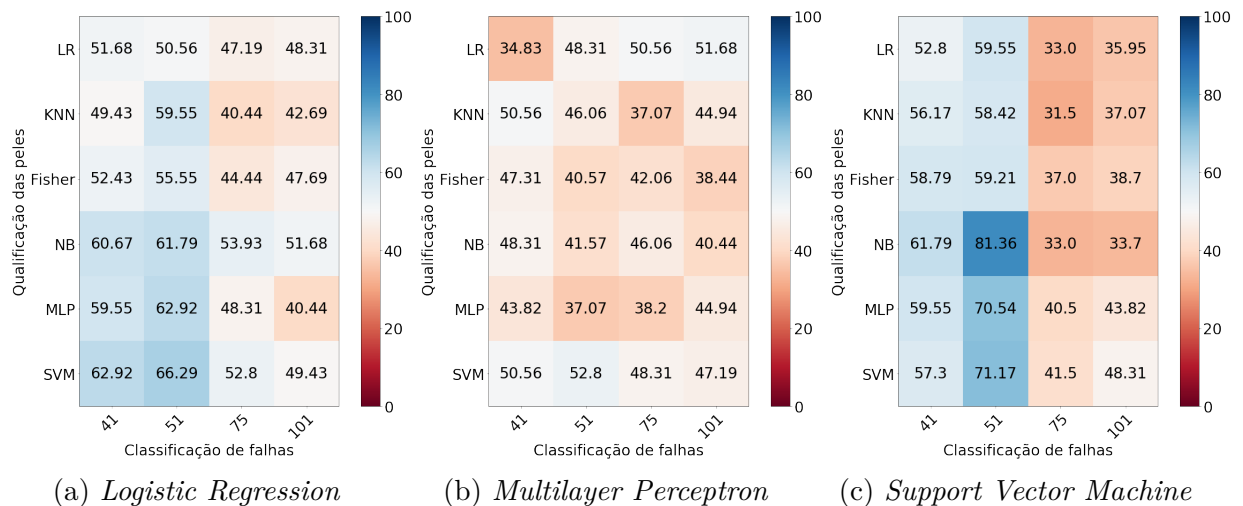
A qualificação das peças de couro ocorre com base nas regiões isentas de falhas e é dividida em duas metodologias. A primeira consiste na busca por regiões sem defeitos com critério de parada na detecção de 80 % dos *pixels* normais e a segunda, ocorre com base em 1, 5 e 10 regiões livres de falhas detectadas.

A partir das detecções das regiões livres de falhas são adquiridos os atributos apresentados na Seção 3.8 e, por fim, são utilizados os classificadores *Logistic Regression*, *K-Nearest Neighbors*, classificador de Fisher, *Naïve Bayes*, *Multilayer Perceptron* e *Support Vector Machine* nos quais os métodos de treinamento e testes são os mesmos apresentados na Seção 3.5.

4.6.1 Resultados obtidos a partir da detecção de 80 % dos *pixels* sem defeitos

Na Figura 38 são apresentados os resultados médios obtidos do processo de teste, com base em 50 execuções, a partir da qualificação das peças de couro caprino e fazendo uso dos melhores resultados adquiridos durante o processo de detecção de falhas.

Figura 38 – Resultados médios obtidos da classificação, com base na detecção de 80% das regiões livres de falhas.



Fonte: Autor.

Como forma de detalhar os resultados médios obtidos, na Tabela 26 são apresentadas as 3 melhores configurações provenientes do processo de qualificação.

Tabela 26 – Melhores resultados médios adquiridos durante o processo de qualificação com base na detecção de 80% dos *pixels* isentos de falhas.

Ordem	Dim. região de interesse	Classificador de falhas	Classificador de peles	Acurácia Média (%) / Desvio Padrão (σ)	
				Treinamento	Teste
1	51 <i>pixels</i>	SVM	NB	90.48 $\pm$ 4.14	81.36 $\pm$ 0.74
2	51 <i>pixels</i>	SVM	SVM	85.62 $\pm$ 4.90	71.17 $\pm$ 4.38
3	51 <i>pixels</i>	SVM	MLP	99.95 $\pm$ 0.33	70.54 $\pm$ 7.83

Como apresentado, percebe-se que o classificador *Naïve Bayes* obtém uma melhor acurácia. Conforme Brownlee (2019), isso se deve pois este classificador não precisa de muitos dados para ter um bom desempenho. Deste modo, precisa apenas de dados

suficientes para entender a relação probabilística de cada atributo isoladamente com a variável de saída.

Nota-se que os melhores resultados médios são obtidos a partir da detecção de falhas com regiões de interesse de 51 x 51 *pixels* e classificador SVM e, para a qualificação das peças de couro caprino, a utilização do classificador *Naïve* Bayes alcançando 90,48 % de acurácia no treinamento e 81,36 % no teste. Deste modo, esta configuração foi a escolhida para as etapas seguintes.

4.6.2 Resultados obtidos a partir da redução do número de regiões de interesse a serem analisados

Com o intuito de obter uma melhor qualificação das peças, os resultados descritos a seguir são baseados na detecção de números fixos de áreas livres de falhas. Assim, o critério de parada consiste na obtenção de 1, 5 ou 10 números de regiões detectadas sem falhas.

Salienta-se que, conforme apresentado na Seção 4.6.1, o melhor processo de detecção de falhas ocorre através de dimensões de regiões de interesse de dimensões iguais a 51 x 51 *pixels* e classificador SVM. Deste modo, na Tabela 27 são apresentados os melhores resultados médios alcançados.

Tabela 27 – Melhores resultados médios obtidos durante a redução do número de regiões de interesse a serem analisadas.

Áreas	Dim. região de interesse	Classificador de falhas	Classificador de peles	Acurácia Média (%) / Desvio Padrão (σ)	
				Treinamento	Teste
1	51 <i>pixels</i>	SVM	NB	95.61 $\pm$ 2.90	86.45 $\pm$ 3.47
5	51 <i>pixels</i>	SVM	NB	95.79 $\pm$ 0.97	89.56 $\pm$ 2.35
10	51 <i>pixels</i>	SVM	NB	95.61 $\pm$ 3.09	83.59 $\pm$ 4.77

Conforme apresentada na Tabela 27, o uso dos atributos adquiridos com base na detecção de 5 regiões livres possuem um melhor resultado na qualificação das peças. Dessa forma, na Tabela 28 é apresentada a matriz de confusão gerada a partir do melhor modelo obtido com base em 50 execuções.

Tabela 28 – Melhor resultado na qualificação de peças de couro caprino utilizando o classificador *Naïve Bayes*.

Real \ Predito	<i>Treinamento</i>							<i>Teste</i>						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	5	1	0	0	0	0	0	2	0	0	0	0	0	0
2	0	6	0	0	0	0	0	0	9	2	0	0	0	0
3	0	0	6	0	0	0	0	0	0	4	1	0	0	0
4	0	0	0	5	1	0	0	0	0	0	12	2	0	0
5	0	0	0	0	6	0	0	0	0	0	0	15	0	0
6	0	0	0	0	0	6	0	0	0	0	0	0	2	1
7	0	0	0	0	0	0	6	0	0	0	0	0	0	39

A seguir, na Tabela 29, são apresentados os demais dados sobre a matriz de confusão apresentada na Tabela 28.

Tabela 29 – Análise da acurácia, precisão, *recall* e F1 do melhor resultado obtido - classificador *Naïve Bayes*.

Classe	<i>Treinamento</i>				<i>Teste</i>			
	Acurácia	Precisão	<i>Recall</i>	F1	Acurácia	Precisão	<i>Recall</i>	F1
1	0,95	1,00	0,83	0,91	0,93	1,00	1,00	1,00
2		0,86	1,00	0,92		1,00	0,82	0,90
3		1,00	1,00	1,00		0,67	0,80	0,73
4		1,00	0,83	0,91		0,92	0,86	0,89
5		0,86	1,00	0,92		0,88	1,00	0,94
6		1,00	1,00	1,00		1,00	0,67	0,80
7		1,00	1,00	1,00		0,98	1,00	0,99

Ao analisar a Tabela 29, torna-se possível verificar que é obtida uma acurácia de 95 % no treinamento e 93 % no teste. Evidencia-se também que a maior dificuldade em classificar é encontrada na classe 3. Contudo, todos os erros de classificação apontam para uma classe próxima à real. Dessa forma, tais problemas encontrados são considerados pequenos, tendo em vista a proximidade da qualificação correta.

5 Conclusões e Sugestões para Trabalhos Futuros

Nesta Dissertação foram apresentados os processos realizados para a construção de um sistema de apoio a decisão para a qualificação de peças de couro caprino *wet blue*.

Salienta-se que o treinamento realizado em um curtume foi de grande importância para o entendimento a cerca da detecção de defeitos e qualificação das peças de couro caprino.

Quanto a elaboração dos conjuntos de imagens, foram obtidas 312 amostras de peças de couro caprino *wet blue*. A partir dessas, com a colaboração de profissionais do curtume, são adquiridas 3919 coordenadas, nos eixos X e Y , de amostras para a detecção de falhas. Com o uso dessas, pode-se adquirir sete conjuntos de imagens fazendo uso de regiões de interesse com dimensões de 11 x 11, 21 x 21, 31 x 31, 41 x 41, 51 x 51, 75 x 75 e 101 x 101 *pixels*.

A partir dos conjuntos de imagens adquiridos pode-se obter 42 conjuntos de dados provindos dos extratores *Gray Level Co-Occurrence Matrix* (GLCM) e *Local Binary Pattern* (LBP). Estes foram submetidos aos processos de treinamento com os classificadores *Single Layer Perceptron*, *Logistic Regression*, *K-Nearest Neighbors* (KNN), Fisher, *Naïve Bayes*, *Multilayer Perceptron* (MLP) e *Support Vector Machine* (SVM). Neste quesito, o modelo que apresentou melhor resultado é composto por regiões de interesse com dimensões de 51 x 51 *pixels*, extrator GLCM e classificador SVM_{RF}, que proporcionou uma acurácia de 96,47 % no treinamento e 96,31 % no teste.

Dados os experimentos, percebeu-se que o modelo de detecção de falhas necessita funcionar de forma ágil na resolução do problema. Tendo em vista tal afirmação, com a adoção de computação paralela, pode-se reduzir o tempo para este processo de, em torno, de 31 para aproximadamente 11 segundos.

Por fim, a partir das detecções de áreas livres de falhas, são obtidos novos atributos e gerados novos conjuntos de dados. Estes são elaborados com base na detecção de apenas 1, 5 e 10 regiões livres de falhas.

Com os conjuntos adquiridos, foi iniciado um novo processo de treinamento com o intuito de qualificar peças de couro caprino. Neste são utilizados os classificadores não lineares apresentados anteriormente. Isto posto, o melhor resultado é obtido com o uso classificador *Naïve Bayes* e busca de 5 maiores regiões isentas de falhas, proporcionando uma acurácia de 95,24 % no treinamento e 93,26 % no teste.

Há nesse cenário um avanço frente às propostas anteriores. Ao considerar os

trabalhos de Amorim et al. (2010), Santos Filho (2013), Aquino (2017), Renato Pereira (2018) e Liong et al. (2019) é perceptível uma melhoria através da elaboração de conjuntos de imagens com um número maior de amostras, o que permitem um melhor processo de treinamento pelos classificadores e nos resultados referentes a detecção de falhas e qualificação das peças de couro.

Nota-se, a cada proposta desenvolvida, um maior nível de maturidade quanto à elaboração de sistemas de apoio à decisão na identificação do nível de qualidade das peças de couro caprino *wet blue*. Isto deve-se aos autores citados, que proporcionaram metodologias factíveis para o melhoramento/implementação desse tipo de solução.

5.1 Trabalhos Futuros

Visando obter conjuntos de dados que representem melhor as amostras utilizadas, pretende-se aplicar técnicas de realce, baseadas em equalização de histograma, correção gamma, entre outras.

Uma outra proposta é utilizar diferentes modelos de cores como *Red, Green and Blue* (RGB), *Hue, Saturation and Intensity* (HUE) e *Cyan, Magenta, Yellow and Black* (CMYK), analisando individualmente cada plano para a elaboração de novos conjuntos de dados e, a partir da aplicação de técnicas para aquisição de atributos como *wavelets*, *Histogram of Oriented Gradients* (HOG), entre outras, elaborar novos conjuntos de dados.

Quanto aos conjuntos de dados elaborados, pretende-se aplicar técnicas que possibilitem o alcance de melhores resultados, como a redução de dimensionalidade, remoção de *outliers*, aquisição de novos atributos e elaboração de novas amostras a partir de técnicas como *Synthetic Minority Oversampling Technique* (SMOTE), *Adaptive Synthetic Sampling* (ADASYN), entre outras.

Por fim, com o objetivo de reduzir o tempo para a detecção de falhas nas peças de couro caprino, pretende-se substituir a técnica de multiprocessamento baseada em processadores computacionais e fazer uso da plataforma de computação que vem ganhando bastante notoriedade devido ao seu desempenho, a *Compute Unified Device Architecture* (CUDA). Esta desenvolvida pela NVIDIA, faz uso de núcleos existentes na *Graphics Processing Unit* (GPU), denominados CUDA cores, que são bem superiores aos processadores contidos no *Central Processing Unit* (CPU) no quesito quantidade. Deste modo, tal tecnologia se torna ideal no sentido de extrair atributos e classificar falhas em diversas regiões de interesse de imagem de uma só vez, reduzindo assim o tempo para a detecção de falhas em uma peça de couro inteira.

Referências

AGARWAL, A. *Building a Logistic Regression in Python*. 2018. Disponível em: <<https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24>>. Acesso em: 08 jan. 2020. Citado 2 vezes nas páginas 117 e 118.

AMORIM, W. P. et al. Attributes reduction applied to leather defects classification. *23rd SIBGRAPI Conference on Graphics, Patterns and Images*, p. 353–359, 2010. Citado 3 vezes nas páginas 36, 84 e 119.

ANDRADE, M. P. de. *Uso de Inteligência Computacional no Reconhecimento Óptico de Caracteres*. Dissertação (Mestrado) — Universidade Cândido Mendes, Campos dos Goytacazes, RJ, Brasil, 2015. Citado na página 124.

AQUINO, J. H. N. de. Extração de características de imagens para classificação da qualidade de couro caprino usando padrão binário local. In: *Dissertação*. Fortaleza, CE, Brasil: Universidade Federal do Ceará, 2017. Citado 3 vezes nas páginas 36, 84 e 119.

ASLAM, M. et al. On the application of automated machine vision for leather defect inspection and grading: A survey. *IEEE Access*, v. 7, p. 176065–176086, 2019. Citado 2 vezes nas páginas 28 e 62.

AZEVEDO, E.; CONCI, A.; LETA, F. *Computação gráfica - volume 2: Teoria e prática*. 1. ed. São Paulo: GEN LTC, 2018. v. 2. (Computação gráfica, v. 2). ISBN 9788535292121. Citado 2 vezes nas páginas 103 e 104.

BACKES, A.; SÁ, J. de M. *Introdução à Visão Computacional Usando MATLAB*. São Paulo: Alta Books, 2019. ISBN 978-85-508-0023-3. Citado 6 vezes nas páginas 99, 106, 107, 108, 110 e 122.

BALDNER, F. et al. *Metrologia por Imagem*. Brasil: Elsevier Editora Ltda, 2017. ISBN 9788535272598. Citado 3 vezes nas páginas 100, 106 e 107.

Banco do Nordeste. *Informe rural ETENE - Setor de peles e de couros de caprinos e de ovinos no Nordeste*. 2012. Citado na página 32.

BARALDI, A.; PARMIGGIANI, F. An investigation of the textural characteristics associated with gray level co-occurrence matrix statistical parameters. 1995. Citado na página 111.

BARELLI, F. *Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV*. Espírito Santo: Casa do Código, 2018. ISBN 9788594188588. Citado na página 104.

BARNEY, B. *Introduction to Parallel Computing*. Lawrence Livermore National Laboratory, 2019. Disponível em: <https://computing.llnl.gov/tutorials/parallel_comp/#Whatis>. Citado 2 vezes nas páginas 127 e 128.

BEYELER, M. *Machine Learning for OpenCV*. Birmingham: Packt Publishing, 2017. ISBN 9781783980291. Citado 2 vezes nas páginas 120 e 121.

- BORTH, M. R.; IACIA, J. C.; PISTORI, H. A visão computacional no agronegócio: Aplicações e direcionamentos. 2014. Citado na página 99.
- BROWNIEE, J. *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*. 2020. Disponível em: <<https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/>>. Acesso em: 26 jul. 2020. Citado na página 54.
- BROWNLEE, J. *Better Naive Bayes: 12 Tips To Get The Most From The Naive Bayes Algorithm*. 2019. Disponível em: <<https://machinelearningmastery.com/better-naive-bayes/>>. Citado na página 79.
- BROWNLEE, J. *Linear Discriminant Analysis for Machine Learning*. 2020. Disponível em: <<https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/>>. Acesso em: 04 abr. 2020. Citado na página 120.
- BRYSON, A. *Applied optimal control: Optimization, estimation and control*. CRC Press, 2018. Citado na página 122.
- BUENO, R. C. Detecção de contornos em imagens de padrões de escoamento bifásico com alta fração de vazio em experimentos de circulação natural com o uso de processamento inteligente. In: *Dissertação*. São Paulo, SP, Brasil: IPEN - Autarquia Associada à Universidade de São Paulo, 2016. Citado na página 104.
- BURKOV, A. *The Hundred-Page Machine Learning Book*. Canada: Andriy Burkov, 2019. ISBN 9781999579517. Citado na página 115.
- CAMELO, H. do N.; LUCIO, P. S.; JUNIOR, J. V. L. Modelagem da velocidade do vento usando metodologias arima, holt-winters e rna na previsão de geração eólica no nordeste brasileiro. *Revista Brasileira de Climatologia*, v. 21, 11 2017. Citado na página 123.
- CAVALCANTE, T. da S. *Método de Superfícies Ativas Usando Local Binary Patterns (LBP) Aplicado na Segmentação de Lobos Pulmonares em Imagens de Tomografia Computadorizada do Tórax*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, CE, Brasil, 2016. Citado na página 113.
- COBRASIL. *A História do Couro*. 2017. Disponível em: <<https://www.cobrasil.com.br/pt/a-historia-do-couro/>>. Acesso em: 20 out. 2018. Citado 3 vezes nas páginas 27, 33 e 34.
- COELHO, L.; RICHERT, W.; BRUCHER, M. *Building Machine Learning Systems with Python: Explore machine learning and deep learning techniques for building intelligent systems using scikit-learn and TensorFlow, 3rd Edition*. Birmingham: Packt Publishing, 2018. ISBN 9781788622226. Citado na página 121.
- COPPIN, B. *Inteligência Artificial*. Rio de Janeiro: LTC, 2010. ISBN 9788521617297. Citado 4 vezes nas páginas 117, 120, 121 e 122.
- COVER, T.; HART, P. Nearest neighbor pattern classification. v. 13, p. 21– 27, 1967. ISSN 0018-9448. Disponível em: <http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1053964>. Citado na página 118.

- DATATURKS. *Understanding SVM's: For Image Classification*. 2018. Disponível em: <<https://medium.com/dataturks/understanding-svms-for-image-classification-cf4f01232700>>. Acesso em: 13 jul. 2019. Citado na página 123.
- DESHPANDE, M. *Perceptrons: The First Neural Networks*. 2017. Disponível em: <<https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/>>. Acesso em: 02 jan. 2020. Citado na página 116.
- Dhaval Dave. *Maximum size of square sub matrix with all 1's in a binary matrix*. 2017. Disponível em: <<https://www.gohired.in/2017/12/12/maximum-size-of-square-sub-matrix-with-all-1-in-a-binary-matrix/>>. Acesso em: 14 jan. 2020. Citado na página 131.
- EMBRAPA. *O agronegócio das peles caprina e ovina*. 2002. Disponível em: <<http://old.cnpqg.embrapa.br/publicacoes/doc/doc127/05agronegocio.html>>. Acesso em: 28 dez. 2019. Citado na página 33.
- EMBRAPA. *Novo Censo Agropecuário mostra crescimento de efetivo de caprinos e ovinos no Nordeste*. 2018. Disponível em: <<https://www.embrapa.br/cim-inteligencia-e-mercado-de-caprinos-e-ovinos/busca-de-noticias/-/noticia/36365362/novo-censo-agropecuário-mostra-crescimento-de-efetivo-de-caprinos-e-ovinos-no-nordeste>>. Acesso em: 28 dez. 2019. Citado 2 vezes nas páginas 31 e 32.
- FISHER, R. A. The statistical utilization of multiple measurements. *Annals of Eugenics*, v. 8, n. 4, p. 376–386, 1938. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1938.tb02189.x>>. Citado na página 120.
- FUKUNAGA, K. *Introduction to Statistical Pattern Recognition. second ed.* Indiana: Academic Press, 1990. Citado na página 121.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3. ed. Sao Paulo: Pearson - Prentice Hall, 2011. Citado 3 vezes nas páginas 99, 102 e 105.
- GVARDINSKAS, M. Combined classification error rate estimator for the fisher linear classifier. *Information Technology And Control*, v. 45, 01 2017. Citado na página 51.
- HAFF, G. *What became of multi-core programming problems*. 2011. Disponível em: <<https://www.cnet.com/news/what-became-of-multi-core-programming-problems/>>. Acesso em: 14 abr. 2019. Citado 2 vezes nas páginas 127 e 128.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS*, SMC-3, n. 6, p. 610–621, 1973. Citado 2 vezes nas páginas 111 e 112.
- HAYKIN, S. *Redes Neurais - 2ed.* Ontário - Canadá: BOOKMAN COMPANHIA ED, 2001. ISBN 9788573077186. Citado 4 vezes nas páginas 52, 116, 122 e 124.
- ITO, R. H.; KIM, H. Y.; SALCEDO, W. J. Classificação de texturas invariante a rotação usando matriz de co-ocorrência. 2019. Citado na página 110.
- KLEIN, B.; BONDENSEO. *k-Nearest-Neighbor Classifier*. 2018. Disponível em: <https://www.python-course.eu/k_nearest_neighbor_classifier.php>. Acesso em: 14 jul. 2019. Citado na página 118.

- KRUSE, R. et al. *Computational Intelligence - A Methodological Introduction*. New York: Springer, 2013. ISBN 978-1-4471-5013-8. Citado na página 115.
- KWAK, C.; VENTURA, J.; TOFANG-SAZI, K. Automated defect inspection and classification of leather fabric. *Intelligent Data Analysis*, IOS Press, v. 5, n. 4, p. 355–370, 12 2001. ISSN 1088-467X. Citado na página 28.
- LANARO, G. *Python High Performance - Second Edition*. 2nd. ed. Birmingham: Packt Publishing, 2017. ISBN 1787282899. Citado na página 127.
- LEAL, L. de S. G.; ROCHA, A. K. P.; JUNIOR, C. J. G. da R. Inovação numa empresa processadora de couro caprino em Pernambuco: limites e potencialidades. 2014. Citado na página 32.
- LEATHER DICTIONARY. *Leather quality*. 2007. Disponível em: <https://www.leather-dictionary.com/index.php/Leather_quality>. Acesso em: 2 abr. 2020. Citado na página 34.
- LIMA, R. L. de. Avaliação do algoritmo svm na detecção de comportamentos suspeitos em cenas de vídeo. In: . Ponta Grossa, PR, Brasil: Universidade Tecnológica Federal do Paraná, 2014. Citado 2 vezes nas páginas 100 e 124.
- LIONG, S. et al. Automatic defect segmentation on leather with deep learning. *CoRR*, abs/1903.12139, 2019. Disponível em: <<http://arxiv.org/abs/1903.12139>>. Citado 2 vezes nas páginas 38 e 84.
- LIU, L. et al. Extended local binary patterns for texture classification. *Image and Vision Computing*, v. 30, n. 2, p. 86 – 99, 2012. Citado na página 109.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Citado na página 115.
- MEL, L. *Kernel SVM in Python*. 2018. Disponível em: <<https://medium.com/data-py-blog/kernel-svm-in-python-a8fae37908b9>>. Acesso em: 16 set. 2019. Citado 2 vezes nas páginas 125 e 126.
- MOREIRA, S. *Rede Neural Perceptron Multicamadas*. 2018. Disponível em: <<https://medium.com/ensina-ai/rede-neural-perceptron-multicamadas-f9de8471f1a9>>. Citado na página 123.
- MOTWANI, K. *Bayesian Inference In Python*. 2019. Disponível em: <<https://stepupanalytics.com/bayesian-inference-in-python/>>. Acesso em: 09 fev. 2020. Citado na página 120.
- OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971 – 987, 2002. Citado 2 vezes nas páginas 50 e 112.
- OTSU, N. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. Disponível em: <<http://dx.doi.org/10.1109/TSMC.1979.4310076>>. Citado 3 vezes nas páginas 37, 55 e 56.

- PALMIERE, S. E. *Rede Perceptron de uma única camada*. 2016. Disponível em: <<https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>>. Acesso em: 16 jul. 2019. Citado na página 116.
- PAZ, S. d. G. E. *Crescimento de Regiões por Agregação de Pixels*. 2017. Disponível em: <https://www.gta.ufrj.br/grad/07_2/eliseu/CrescimentodeRegiesporAgregaodePixels.html>. Acesso em: 25 nov. 2018. Citado na página 105.
- PIERFEDERICI, F. *Distributed Computing with Python*. Birmingham: Packt Publishing, 2016. ISBN 978-1-78588-969-1. Citado na página 127.
- PIETIKAINEN, M. et al. Computer vision using local binary patterns. *Springer*, 2011. Citado na página 113.
- RAMALHO, G. et al. Rotation-invariant feature extraction using a structural co-occurrence matrix. v. 94, 08 2016. Citado na página 111.
- RASCHKA, S. *Python Machine Learning*. Birmingham: Packt Publishing, 2015. ISBN 1783555130. Citado na página 117.
- Renato Pereira, F. *Qualificação de Amostras de Couro Caprino Utilizando Visão Computacional e Inteligência Computacional*. Dissertação (Mestrado) — Instituto Federal do Ceará, Fortaleza, CE, Brasil, 2018. Citado 8 vezes nas páginas 37, 38, 45, 55, 58, 60, 84 e 119.
- RETHANA, M. *Bayesian Statistics and Naive Bayes Classifier Refresher*. 2018. Disponível em: <<https://medium.com/@mark.rethana/bayesian-statistics-and-naive-bayes-classifier-33b735ad7b16>>. Acesso em: 09 fev. 2020. Citado na página 121.
- Roberto e Souza, M. Uma análise de diferentes abordagens para o reconhecimento off-line de estilos de escrita em documentos manuscritos. In: *TCC*. Campos Mourão, PR, Brasil: Universidade Tecnológica Federal do Paraná, 2018. Citado na página 112.
- ROSEBROCK, A. *Local Binary Patterns with Python and OpenCV*. 2015. Disponível em: <<https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>>. Acesso em: 21 abr. 2019. Citado na página 112.
- ROSENBLATT, F. The perceptron: a theory of statistical separability in cognitive systems (project para). Cornell Aeronautical Laboratory, 1958. Citado 2 vezes nas páginas 115 e 117.
- SANTOS, A. A. M. A. et al. *Panorama do setor de couro no Brasil*. Rio de Janeiro: Banco Nacional de Desenvolvimento Econômico e Social, 2018. Citado na página 31.
- Santos Filho, E. *Um Sistema de Visão Computacional para Classificação de Couro Caprino*. Dissertação (Mestrado) — Universidade Federal do Ceará (UFC), Fortaleza, CE, Brasil, 2013. Citado 5 vezes nas páginas 34, 35, 36, 46 e 84.
- SANTOS FILHO, E. Queiroz dos; BARRETO, G.; DANIEL SANTOS, J. Estudo comparativo de métodos de extração de características para classificação da qualidade de peles de caprinos com opção de rejeição. In: . Fortaleza-CE: XI Simpósio Brasileiro de Automação Inteligente, 2013. ISBN 2358-4483. Citado na página 27.

- SARETTA, C. B. Tributação do couro wet blue e suas implicações ao setor coureiro. *Leituras de Economia Política*, v. 10, n. 10, 2012. Citado na página 27.
- SCHWARTZ, W. R.; PEDRINI, H. Método para classificação de imagens baseada em matrizes de co-ocorrência utilizando características de textura. *III Colóquio Brasileiro de Ciências Geodésicas*, Universidade Federal do Pará, 2003. Citado na página 109.
- SHRIMALI, K. R. *SVM using Scikit-Learn in Python*. 2018. Disponível em: <<https://www.learnopencv.com/svm-using-scikit-learn-in-python/>>. Acesso em: 16 set. 2019. Citado na página 125.
- SHUNG, K. P. *Accuracy, Precision, Recall or F1?* 2018. Disponível em: <<https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>>. Acesso em: 26 jul. 2020. Citado na página 54.
- SILVA, I.; SPATTI, D.; FLAUZINO, R. *Redes Neurais Artificiais para Engenharia e Ciências Aplicadas*. 2. ed. ARTLIBER, 2019. ISBN 9788588098534. Disponível em: <<https://books.google.com.br/books?id=w2VHbwAACAAJ>>. Citado na página 52.
- SILVA, R. R. V. e. *Detecção Automática de Células Cervicais com Base em Atributos Radiais*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, CE, Brasil, 2018. Citado na página 109.
- SOBEL, I. Camera models and machine perception. Department of Electrical Engineering, Stanford University., 1970. Citado na página 104.
- TARANTOLA, A. *Como o couro está matando lentamente as pessoas e lugares que o produzem*. 2014. Disponível em: <<https://gizmodo.uol.com.br/couro-giz-explica/>>. Citado na página 34.
- VAPNIK, V. N. The nature of statistical learning theory. *Spring - Verlag New York*, 1995. Citado na página 123.
- YADAV, A. *Support Vector Machines (SVM)*. 2018. Disponível em: <<https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589>>. Acesso em: 17 set. 2019. Citado na página 126.
- ZANOTTA, D.; FERREIRA, M.; ZORTEA, M. *Processamento de imagens de satélite*. São Paulo: Oficina de Textos, 2019. ISBN 9788579753176. Citado 3 vezes nas páginas 99, 105 e 106.

Apêndices

APÊNDICE A – Qualificações definidas pelos especialistas para elaboração do conjunto de imagens

Tabela 30 – Qualificações definidas pelos classificadores.

N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2
1	1	1	41	2	1	81	3	3	121	4	6
2	1	1	42	2	1	82	3	6	122	4	4
3	1	2	43	2	2	83	3	5	123	4	4
4	1	5	44	2	2	84	3	5	124	4	5
5	1	5	45	2	2	85	3	3	125	4	4
6	1	2	46	2	3	86	3	5	126	4	5
7	1	1	47	2	3	87	3	6	127	4	4
8	1	1	48	2	4	88	3	5	128	4	5
9	1	1	49	2	1	89	3	3	129	4	4
10	1	2	50	2	4	90	3	3	130	4	5
11	1	4	51	2	5	91	3	3	131	4	6
12	1	1	52	2	2	92	3	2	132	4	4
13	1	3	53	2	2	93	3	3	133	4	5
14	1	1	54	2	4	94	3	5	134	4	6
15	1	2	55	2	2	95	3	6	135	4	6
16	1	1	56	2	2	96	3	5	136	4	3
17	1	2	57	2	3	97	3	2	137	4	4
18	1	5	58	2	2	98	3	5	138	4	5
19	1	4	59	2	4	99	3	5	139	4	4
20	1	2	60	2	2	100	3	6	140	4	4
21	1	3	61	2	5	101	3	5	141	4	3
22	2	1	62	2	4	102	3	5	142	4	4
23	2	5	63	2	2	103	3	1	143	4	6
24	2	2	64	2	1	104	3	5	144	4	3
25	2	2	65	2	1	105	3	6	145	4	5
26	2	5	66	3	6	106	3	4	146	4	4
27	2	3	67	3	3	107	3	6	147	4	4
28	2	1	68	3	5	108	3	5	148	4	5
29	2	1	69	3	7	109	3	6	149	4	5
30	2	3	70	3	1	110	3	6	150	4	5
31	2	5	71	3	5	111	3	6	151	4	4
32	2	1	72	3	6	112	3	3	152	4	5
33	2	3	73	3	2	113	3	2	153	4	5
34	2	2	74	3	3	114	4	4	154	4	4
35	2	2	75	3	3	115	4	5	155	4	4
36	2	3	76	3	6	116	4	4	156	4	2
37	2	6	77	3	5	117	4	6	157	4	5
38	2	2	78	3	5	118	4	4	158	4	2
39	2	2	79	3	3	119	4	5	159	4	4
40	2	2	80	3	6	120	4	5	160	4	6

Tabela 31 – Qualificações definidas pelos classificadores - continuação.

N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2	N	Esp. 1	Esp. 2
161	4	4	201	5	5	241	6	7	281	7	7
162	4	5	202	5	6	242	6	6	282	7	7
163	4	5	203	5	5	243	6	7	283	7	7
164	5	5	204	5	5	244	6	6	284	7	6
165	5	7	205	5	6	245	6	5	285	7	7
166	5	5	206	5	6	246	6	6	286	7	7
167	5	5	207	5	7	247	6	7	287	7	7
168	5	6	208	5	5	248	6	7	288	7	7
169	5	7	209	5	6	249	6	7	289	7	7
170	5	5	210	5	6	250	6	7	290	7	7
171	5	6	211	5	5	251	6	7	291	7	7
172	5	5	212	5	6	252	6	7	292	7	7
173	5	6	213	6	7	253	6	7	293	7	7
174	5	6	214	6	7	254	6	7	294	7	7
175	5	6	215	6	7	255	6	7	295	7	7
176	5	5	216	6	7	256	6	7	296	7	7
177	5	7	217	6	7	257	6	7	297	7	7
178	5	6	218	6	7	258	6	7	298	7	7
179	5	6	219	6	6	259	6	7	299	7	6
180	5	4	220	6	7	260	6	5	300	7	7
181	5	5	221	6	7	261	6	6	301	7	7
182	5	5	222	6	4	262	6	7	302	7	6
183	5	7	223	6	6	263	7	7	303	7	7
184	5	5	224	6	6	264	7	7	304	7	7
185	5	5	225	6	7	265	7	7	305	7	7
186	5	7	226	6	6	266	7	6	306	7	7
187	5	5	227	6	7	267	7	7	307	7	7
188	5	6	228	6	7	268	7	6	308	7	7
189	5	5	229	6	7	269	7	7	309	7	7
190	5	5	230	6	7	270	7	7	310	7	7
191	5	6	231	6	7	271	7	7	311	7	7
192	5	6	232	6	5	272	7	7	312	7	7
193	5	6	233	6	7	273	7	7			
194	5	5	234	6	6	274	7	7			
195	5	6	235	6	7	275	7	7			
196	5	6	236	6	7	276	7	7			
197	5	5	237	6	7	277	7	7			
198	5	5	238	6	7	278	7	7			
199	5	6	239	6	7	279	7	7			
200	5	6	240	6	7	280	7	7			

Anexos

ANEXO A – Visão Computacional

Segundo Backes e Sá (2019), Visão Computacional pode ser definida como a área de estudo que tenta repassar para máquinas a incrível capacidade da visão. Ela consiste em captar imagens, melhorá-las, a partir da retirada de ruídos, separação de regiões ou objetos de interesse de uma cena, extração de várias informações dependendo da imagem analisada, como, por exemplo, forma, cor e textura, e, finalmente, relacionando as imagens com outras vistas previamente.

Segundo Borth, Iacia e Pistori (2014), a visão computacional é uma área de pesquisa que pode incluir métodos de aquisição de imagens, pré-processamento, segmentação, extração de atributos ou características e reconhecimento de padrões.

De acordo com Gonzalez e Woods (2011), um sistema de visão computacional é composto pelas etapas de aquisição, segmentação, extração de características e classificação. Desta forma, as soluções da visão computacional são baseadas em técnicas de processamento e análise de imagens, as quais permitem extrair informações a partir delas.

A seguir são apresentados os métodos de processamento de imagens utilizados neste trabalho.

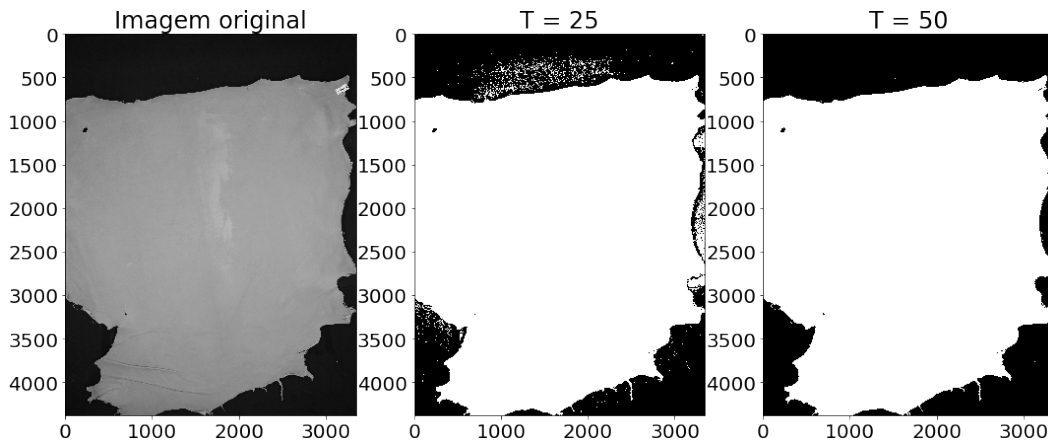
A.1 Limiarização (*thresholding*)

De acordo com Zanotta, Ferreira e Zortea (2019) a limiarização é considerada como o mais simples método para a segmentação de imagens. Desta forma, como apresentada na Equação A.1, dada uma imagem em tons de cinza na qual pode ser descrita por $f(x; y)$, e $(x; y)$ corresponde à posição de um certo *pixel* com nível de cinza descrito pela função f , a limiarização gera uma imagem binária (um ou zero), separando os *pixels* com intensidades superiores (objeto) e inferiores (plano de fundo), ou vice-versa, a um determinado limiar T .

$$f(x; y) = \begin{cases} 1, & \text{se } f(x; y) > T \\ 0, & \text{se } f(x; y) \leq T \end{cases} \quad (\text{A.1})$$

A ideia de limiarização representa a separação de regiões. Deste modo, conforme apresentado na Figura 39, é apresentada a limiarização em uma imagem digital de uma amostra de couro caprino. Nota-se que na subfigura à esquerda (a) é apresentada a imagem original, ao centro (b), a imagem original após um processo de limiarização com valor do parâmetro $T = 25$, e à direita, $T = 50$.

Figura 39 – Limiarização com diversos parâmetros distintos para T .



Fonte: Autor.

Percebe-se que ao inserir o valor 25 ao parâmetro T no problema apresentado, a imagem resultante apresenta diversos ruídos no plano de fundo da mesma. Em contrapartida, ao inserir o valor 50, a limiarização ocorre de maneira satisfatória. Isso demonstra a importância da escolha correta para o parâmetro T .

A.2 Filtros Passa-baixa

Os filtros passa-baixa eliminam as frequências altas sem atuar nas frequências baixas, desta forma, o resultado obtido desses métodos é a suavização de imagens.

Estes filtros tendem a minimizar ruídos, porém apresentam efeito de borramento.

Segundo Lima (2014), diversos filtros de vizinhança, ou seja, que utilizam *pixels* vizinhos, atuam como filtros passa-baixa. Os filtros da média e mediana fazem parte desse grupo e são utilizados neste trabalho.

A.2.1 Filtro da Média

O filtro da média, assim como os demais filtros lineares de suavização, têm por objetivo exibir partes de interesse em uma imagem. É possível afirmar que, basicamente, utiliza a média aritmética da vizinhança e acrescenta este valor ao centro da matriz.

De acordo com Baldner et al. (2017), o filtro da média é um exemplo de filtro linear de passa-baixa. As máscaras utilizadas na convolução resultam na alteração do *pixel* da imagem pela média dos *pixels* vizinhos, podendo variar de acordo com o tamanho da máscara (*kernel*). Esta é formada por uma matriz cujos componentes são iguais a $\frac{1}{M^2}$, em que M representa a quantidade de linhas ou colunas.

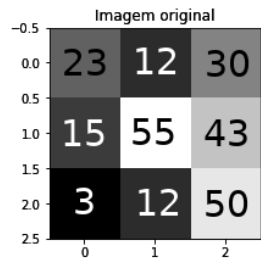
De uma forma mais simples, como apresentado na Equação A.2, a máscara pode ser representada por uma matriz de 1's, onde cada um desses valores são divididos pela

quantidade de posições dela. Desta forma, uma matriz 3 x 3, possui o valor 1 dividido por 9 (total de elementos) em cada posição.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (\text{A.2})$$

Considerando uma matriz 3 x 3 como um fragmento de imagem qualquer, representada na Figura 40, para a aplicação do filtro citado, basta obter a média dos elementos dessa matriz.

Figura 40 – Fragmento de imagem.



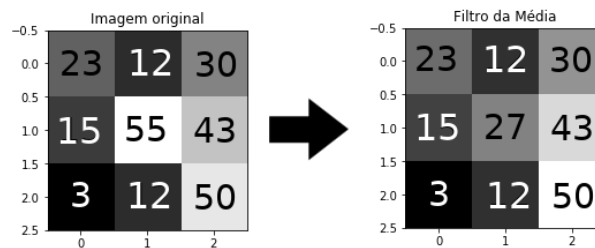
Fonte: Autor.

Conforme pode ser visto na Equação A.3, o valor resultante é 27.

$$\frac{23 + 12 + 30 + 15 + 55 + 43 + 3 + 12 + 50}{9} = 27 \quad (\text{A.3})$$

Nestes termos, conforme apresentado na Figura 41, o valor obtido será inserido no *pixel* central do fragmento da imagem.

Figura 41 – Filtro da média aplicado em um fragmento de imagem.

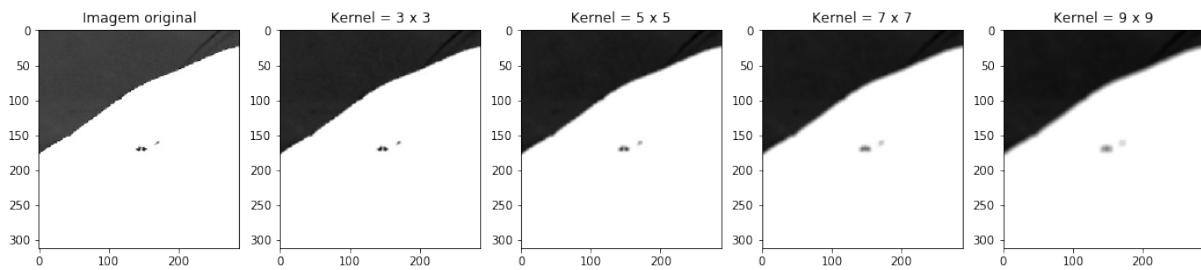


Fonte: Autor.

Salienta-se que a máscara deve calcular todos os *pixels* existentes na imagem, ou seja, percorrendo-a por completo, este processo é denominado como convolução.

Na Figura 42 é apresentado o processo de aplicação do filtro da média em um fragmento de imagem relacionada a uma amostra de couro caprino com diversos tamanhos de *kernel's* diferentes. Nota-se que quanto maior o tamanho do *kernel*, mais a imagem tende a apresentar borramentos.

Figura 42 – Filtro da média aplicado em um fragmento de imagem relacionado à borda de uma pele.



Fonte: Autor.

É possível perceber que o filtro da média é incapaz de preservar bordas. O uso de um *kernel* grande tende a tornar a imagem cada vez mais borrada. No entanto, este filtro obtém bons resultados quanto à redução de ruídos pós-limiarização.

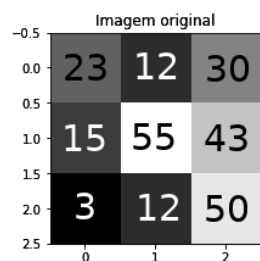
A.2.2 Filtro da Mediana

O filtro da mediana faz parte do grupo de filtros passa-baixa e também apresenta borramento na imagem, porém de uma forma menos agressiva.

Segundo Gonzalez e Woods (2011), o filtro da mediana é um filtro espacial não linear, cuja resposta se baseia na ordenação (classificação) dos *pixels* contidos na área da imagem coberta pelo filtro, substituindo o valor do *pixel* central pelo valor determinado do resultado da classificação.

Considerando o fragmento de imagem apresentado na Figura 43, o filtro da mediana consiste em obter o valor referente à mediana dos elementos contidos na matriz e inserir o mesmo no *pixel* central.

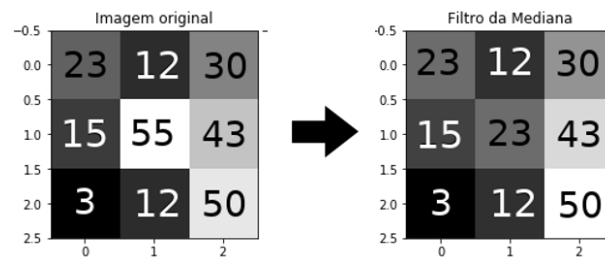
Figura 43 – Fragmento de imagem.



Fonte: Autor.

Desta forma, os elementos do fragmento citado, ao serem ordenados, resultarão em: 3, 12, 12, 15, 23, 30, 43, 50 e 55, na qual a mediana consiste no valor 23. Na Figura 44 é apresentado o resultado final da aplicação do filtro.

Figura 44 – Filtro da média aplicado em um fragmento de imagem.

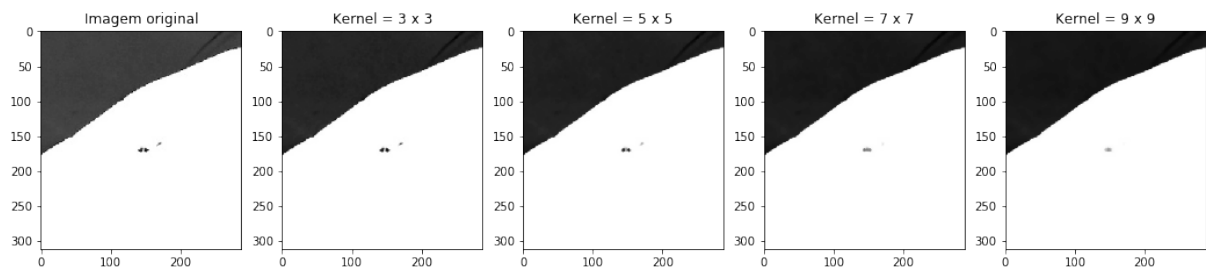


Fonte: Autor.

Como visto, ao obter um conjunto de *pixels* de uma imagem, de acordo com o tamanho do *kernel*, os valores contidos nesse grupo são ordenados e, posteriormente, o *pixel* central é substituído pelo valor da mediana dos mesmos.

Conforme apresentado na Figura 45, nota-se que ocorre um borramento na imagem e que o mesmo se intensifica à medida que o tamanho do *kernel* aumenta, porém, este método apresenta um menor teor de borramentos que o filtro da média.

Figura 45 – Filtro da mediana aplicado em um fragmento de imagem relacionado à borda de uma pele.



Fonte: Autor.

Durante as implementações, torna-se claro que o filtro da mediana é um método apropriado para redução de ruídos pós-limiarização e remoção de manchas em peças de couro caprino não consideradas como falhas.

A.3 Filtros Passa-alta

Diferentes dos filtros passa-baixa, citados na Subseção A.2, os filtros passa-alta eliminam as frequências baixas e não atuam nas frequências altas, desta forma, de acordo com Azevedo, Conci e Leta (2018), estes filtros fazem com que detalhes finos da imagem sejam enfatizados.

A.3.1 Operadores de Sobel

Segundo Bueno (2016), os operadores de Sobel (1970) são métodos clássicos na detecção de bordas que utilizam a primeira derivada com o objetivo de identificar as mudanças locais significativas nos níveis de cinza.

De acordo com Barelli (2018), o operador de Sobel, também nomeado de filtro de Sobel, é uma operação usada para realçar contornos em imagens. Esse filtro não linear realça linhas verticais e horizontais mais escuras que o fundo, sem realçar pontos isolados.

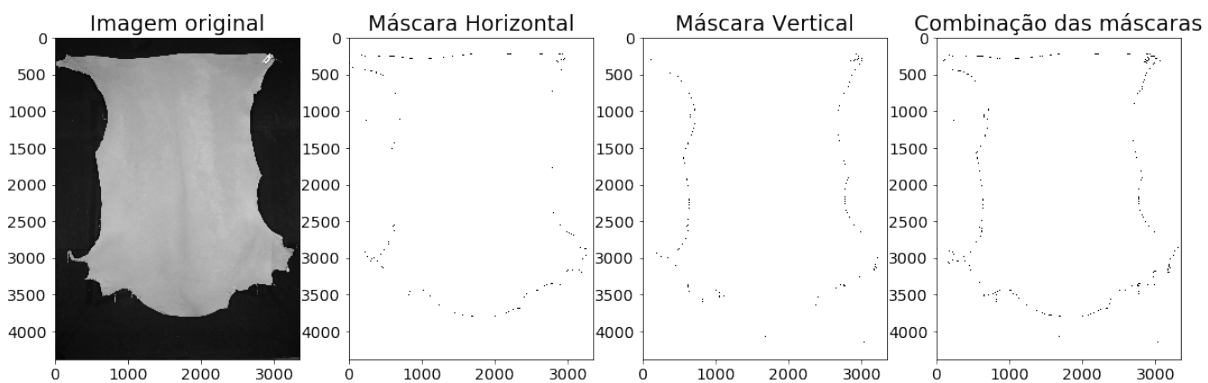
O filtro de Sobel pode ser representado por duas máscaras. A primeira, apresentada na Equação A.4, acentua variações de linhas no sentido horizontal e a segunda, Equação A.5, no sentido vertical. Após a aplicação das duas máscaras, o resultado é combinado obtendo assim, a borda de um objeto de interesse nos sentidos horizontal e vertical.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (\text{A.4})$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (\text{A.5})$$

Na figura 46, é apresentada uma amostra de couro caprino, a aplicação das máscaras horizontal e vertical e a combinação de ambas.

Figura 46 – Aplicação dos operadores de Sobel.



Fonte: Autor.

De acordo com Azevedo, Conci e Leta (2018), os operadores de Sobel possuem a vantagem de prover a diferenciação e a suavização da imagem ao mesmo tempo. Ou seja, incluem também, o efeito de suavização como atrativo para reduzir o impacto dos ruídos e possui a propriedade de realçar linhas verticais e horizontais sem realçar pontos isolados.

Segundo Gonzalez e Woods (2011), um dos fatores das máscaras de Sobel serem bastantes utilizadas é devido ao fato de que as mesmas apresentam melhor supressão de ruído (suavização).

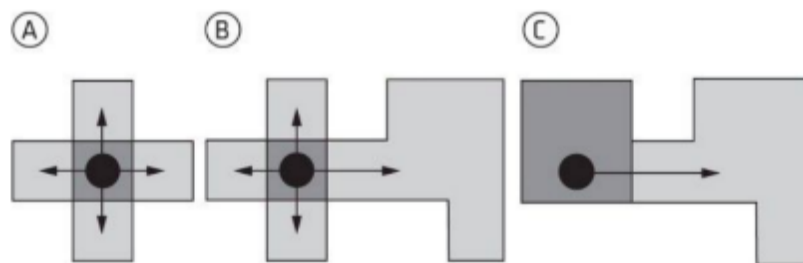
Percebe-se que o filtro de Sobel é bastante útil quando se deseja obter o contorno de objetos para posterior extração dos mesmos. Neste trabalho, é utilizado com o intuito de remoção do plano de fundo, ou seja, obtenção de um objeto de interesse, a peça de couro caprino.

A.4 Crescimento de Regiões

O crescimento de regiões funciona como um processo iterativo, pois examina os *pixels* vizinhos dos *pixels* sementes e determina se deverão ou não serem adicionados a uma determinada região.

Conforme apresentado na Figura 47, o processo é semelhante a uma limiarização e se inicia através de um conjunto de *pixels*. Para cada semente é avaliado o valor de seus vizinhos e em seguida, a agregação de regiões é elaborada de acordo com a sua similaridade.

Figura 47 – Processo de crescimento de região.



Fonte: Zanotta, Ferreira e Zortea (2019).

Segundo Paz (2017), o crescimento das regiões em si, constitui-se em fazer o agrupamento de *pixels* por similaridade em algum critério como intensidade de cinza, textura, etc.

Desta forma, pode-se afirmar que a técnica de segmentação através do crescimento de regiões ocorre em três etapas: escolha dos *pixels* sementes (1), decisão do limiar (2) e crescimento de regiões (3).

Na Figura 48, na subfigura à esquerda, a região na cor vermelha representa o local da escolha do *pixel* semente. Percebe-se que nas imagens posteriores, da esquerda para a direita, ocorre o processo de crescimento de regiões brancas. Como a imagem é binária, todos os *pixels* que contém o mesmo valor são adicionados, obtendo como resultado final a imagem à direita.

Figura 48 – Processo de crescimento de região em uma imagem binária.



Fonte: Autor.

De acordo com Zanotta, Ferreira e Zortea (2019), à medida que a região vai crescendo, novos *pixels* vão sendo analisados e adicionados ao segmento. Em estágios avançados do processo, pode haver junção entre regiões adjacentes, formando segmentos ainda maiores. O resultado produz um conjunto de aglomerados de *pixels*, dando origem aos segmentos.

Conforme destacado na Figura 48, este tipo de segmentação é se torna útil na extração de regiões de interesse. No exemplo citado, nota-se que com este método é possível destacar a peça de couro caprino do plano de fundo.

Em imagens binárias, o processo costuma ocorrer de forma rápida e possuir bons resultados, pois, de certo modo, não há uma preocupação com o valor do *threshold* a ser inserido e requer pouco poder de processamento.

A.5 Operações Morfológicas

Segundo Backes e Sá (2019), uma operação morfológica pode ser entendida como uma transformação na imagem, a qual é realizada utilizando uma segunda imagem menor e cuja estrutura e geometria sejam conhecidas. A essa segunda imagem é dada a denominação de elemento estruturante. Por ter uma estrutura e geometria conhecidas, o elemento estruturante é utilizado para ressaltar ou remover aspectos específicos dos objetos da imagem maior, no caso, a imagem transformada.

As operações morfológicas baseiam-se na teoria de conjuntos. Como citado, utilizam um elemento estruturante, aplicam-no a uma imagem e resultam uma outra imagem como saída. Algumas operações morfológicas são erosão e dilatação, servem para remoção de ruídos, isolamento de elementos individuais, descoberta de buracos em imagens, entre outros.

A.5.1 Erosão

De acordo com Baldner et al. (2017), a erosão provoca o estreitamento na imagem e pode ser utilizada na eliminação de ruídos em imagens binarizadas e na separação de

regiões.

Desta forma, pode-se afirmar que a técnica de erosão consiste em alterar o valor do *pixel* analisado para um valor idêntico ao *pixel* de menor intensidade da imagem original. Na Figura 49 é apresentado o processo de erosão e seu elemento estruturante.

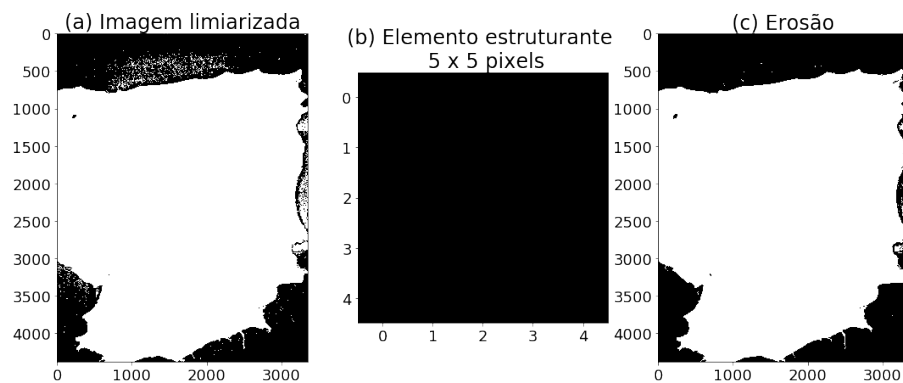
Figura 49 – Exemplo da aplicação da erosão.



Fonte: Backes e Sá (2019).

Desta forma, na prática é possível obter resultados semelhantes à Figura 50. Pois, conforme apresentado, o método possui bastante eficácia na redução de regiões brilhantes da imagem.

Figura 50 – Aplicação do método de erosão em uma imagem limiarizada de amostra de couro caprino.



Fonte: Autor.

A.5.2 Dilatação

Segundo Baldner et al. (2017), a dilatação é uma operação morfológica que consiste na transformação da tonalidade de um *pixel* para um dado padrão (preto ou branco). A operação é realizada entre uma imagem e um elemento estruturante, este, possui a função de determinar o formato da área analisada para a dilatação e o elemento a ser dilatado.

Conforme apresentado na Figura 51, a dilatação utiliza a mesma ideia da erosão, a diferença fica por parte do valor calculado que passa a ser o máximo, ou seja, é uma operação de maximização, aumentando assim, as regiões brilhantes da imagem.

Figura 51 – Exemplo da aplicação da dilatação.

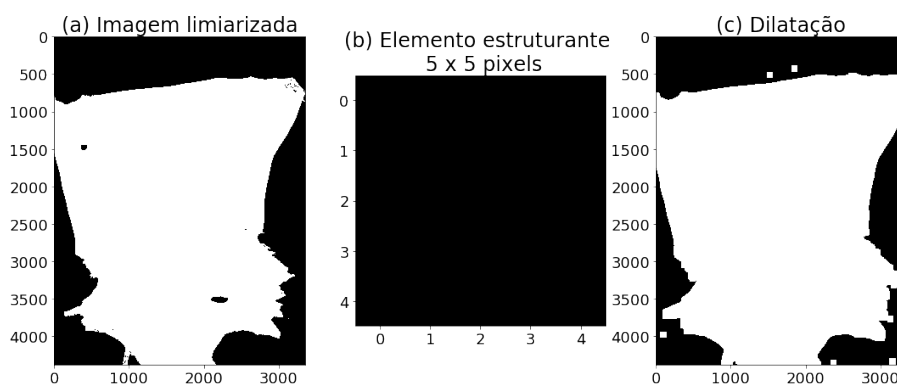


Fonte: Backes e Sá (2019).

Como visto, a partir da dilatação é possível obter um alargamento entre os objetos da imagem, isto se torna bastante útil quando é necessário unir regiões que possam ser separadas durante o processo de segmentação.

Na Figura 52, é apresentado um processo de remoção de buracos contidos na imagem limiarizada de amostra de couro caprino. Percebe-se que ao aplicar o método de dilatação, regiões que possuem buracos são substituídas por *pixels* brancos. Esta aplicação é de suma importância quando se pretende obter a área completa de uma peça de couro caprino que possui buracos.

Figura 52 – Aplicação de dilatação e seus elementos estruturantes em um fragmento de imagem relacionado à borda de uma peça de couro.



Fonte: Autor.

Segundo Backes e Sá (2019), a dilatação pode ser combinada com outros métodos de morfologia matemática para gerar outras operações mais complexas. Neste sentido, conforme apresentado na Figura 52, a mesma possui o intuito de excluir os buracos contidos na amostra de couro caprino limiarizada. Contudo, é possível verificar na Figura 52 (c), que ruídos surgiram a partir desta implementação. Esta problemática pode ser facilmente resolvida a partir da técnica de erosão.

A.6 Extratores de Características

Conforme Liu et al. (2012), muitos problemas de visão computacional podem ser solucionados com métodos de reconhecimento de textura. Situações como análise de imagens de satélites, inspeção industrial, diagnóstico médico por imagens, verificação da qualidade de materiais e reconhecimento de faces, exemplificam áreas em que podem ser aplicadas técnicas de classificação de textura. Apesar de parecer simples para os padrões atuais de computação, essa tarefa tem se mostrado desafiadora e continua em desenvolvimento.

A seguir, serão explanados dois extratores de características baseados em texturas *Gray-Level Co-Occurrence Matrix* (GLCM) e *Local Binary Patterns* (LBP) que são utilizados neste trabalho.

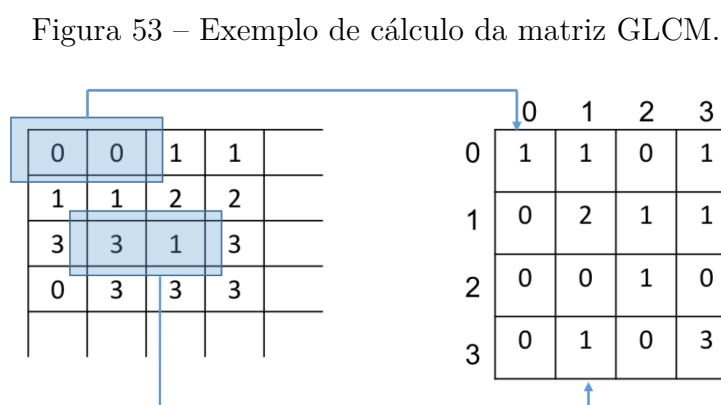
A.6.1 *Gray-Level Co-Occurrence Matrix* (GLCM)

Segundo Schwartz e Pedrini (2003), a matriz de co-ocorrência é uma tabulação de quantas combinações diferentes de valores de intensidade dos *pixels* (níveis de cinza) ocorrem em uma imagem. A ideia principal da matriz de co-ocorrência é descrever textura através de um conjunto de características para as ocorrências de cada nível de cinza nos *pixels* da imagem considerando múltiplas direções.

A matriz GLCM, é voltada para uma série de cálculos de textura de segunda ordem, ou seja, considera a relação entre grupo de dois *pixels* geralmente vizinhos.

Para a obtenção da GLCM, é analisada a imagem original e em seguida verificada a relação espacial entre dois *pixels*. Desta forma, cada elemento contido na segunda matriz é a quantidade de vezes que o *pixel* com valor i ocorreu na relação para o j .

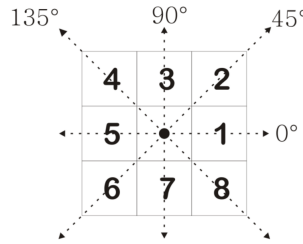
Na Figura 53, é apresentado um exemplo de cálculo para a montagem da matriz de co-ocorrências. A subfigura à esquerda representa um fragmento de imagem e, à direita, a matriz de co-ocorrências.



Fonte: Silva (2018).

De acordo com Backes e Sá (2019), o método de matrizes de co-ocorrência é bastante simples e pode ser definido como: dada uma imagem em níveis de cinza $f(x, y)$ com L valores discretos de intensidade, a matriz de co-ocorrência $C_{d\theta}(i, j)$ é $L \times L$ e em cada posição de i e j recebe a quantidade de vezes que os níveis de cinza i e j da imagem $f(x, y)$ estão presentes em uma distância d e uma orientação θ (Figura 54).

Figura 54 – Ângulos utilizados para o cálculo da matriz de co-ocorrência



Fonte: Ito, Kim e Salcedo (2019).

Salienta-se que durante a construção da GLCM há dois modos de contagem de co-ocorrências, a simétrica e a assimétrica.

A simétrica distingue-se por sua contagem ocorrer em ambos os sentidos, ou seja, como por exemplo, da direita para à esquerda e vice-versa. Em contrapartida, a assimétrica ocorre em apenas um sentido.

Na Figura 55 é apresentado um fragmento de imagem (a), a construção de uma matriz de co-ocorrência - $C_{d\theta}$ (b), $C_{d\theta}$ simétrica com parâmetros $d = 1$ e $\theta = 90^\circ$ (c) e $C_{d\theta}$ assimétrica com $d = 1$ e $\theta = 0^\circ$ (d).

Figura 55 – (a) Fragmento de imagem, (b) construção de uma matriz de co-ocorrência ($C_{d\theta}$), (c) $C_{d\theta}$ simétrica ($d = 1$, $\theta = 90^\circ$) (d) $C_{d\theta}$ assimétrica ($d = 1$, $\theta = 0^\circ$)

0	0	0
2	1	2
2	0	3

(a) fragmento de imagem

i/j	0	1	2	3
0	C(0,0)	C(0,1)	C(0,2)	C(0,3)
1	C(1,0)	C(1,1)	C(1,2)	C(1,3)
2	C(2,0)	C(2,1)	C(2,2)	C(2,3)
3	C(3,0)	C(3,1)	C(3,2)	C(3,3)

(b) construção de uma $C_{d\theta}$

0	2	2	0
2	0	0	0
2	0	2	1
0	0	1	0

(c) $C_{d\theta}$ simétrica ($d = 1$, $\theta = 90^\circ$)

2	0	0	1
0	0	1	0
1	1	0	0
0	0	0	0

(d) $C_{d\theta}$ assimétrica ($d = 1$, $\theta = 0^\circ$)

Fonte: Adaptada de Backes e Sá (2019).

Com a GLCM obtida, é iniciado o processo de obtenção de atributos da mesma. Haralick, Shanmugam e Dinstein (1973) propõem 14 medidas estatísticas. No entanto, Baraldi e Parmiggiani (1995), informa que apenas seis delas são de fato relevantes, sendo descritas a seguir.

Vale ressaltar que $p(i, j)$ representa a matriz de co-ocorrência, N_g , a quantidade de valores discretos da imagem original, $\mu = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i p(i, j)$, μ_x e μ_y os valores médios e σ_x e σ_y os desvios padrões.

1. Energia ou Segundo Momento Angular: é uma medida que determina a uniformidade da textura. A Energia é definida conforme a Equação A.6.

$$f_{ene} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)^2 \quad (\text{A.6})$$

2. Entropia: representada na Equação A.7, calcula o grau de desordem de uma imagem, ou seja, a presença de transições abruptas dos níveis de cinza na imagem.

$$f_{ent} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p(i, j)) \quad (\text{A.7})$$

3. Contraste: conforme mostrado na Equação A.8, representa a variação de níveis de cinza de uma imagem, ou seja, a diferença de intensidade de um pixel e seus vizinhos.

$$f_{con} = - \sum_{n=0}^{N_g-1} n^2 \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right)_{|i-j|=n} \quad (\text{A.8})$$

4. Variância: apresentada na Equação A.9, mensura a heterogeneidade de uma imagem, retorna valores altos se os tons de cinza forem diferentes do nível de cinza médio.

$$f_{var} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 p(i, j) \quad (\text{A.9})$$

5. Homogeneidade: apresentada na Equação A.10, mede o quanto a distribuição dos elementos da GLCM está próxima da diagonal principal (RAMALHO et al., 2016).

$$f_{hom} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} p(i, j) \quad (\text{A.10})$$

6. Correlação: como apresentada na Equação A.11, mensura a dependência linear dos níveis de cinza entre os *pixels*.

$$f_{cor} = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (\text{A.11})$$

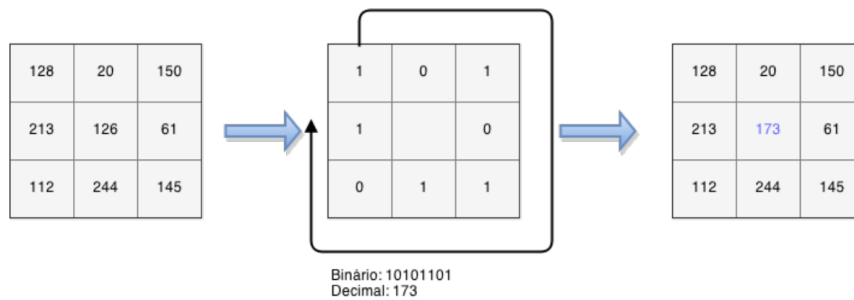
Como explanado, através da GLCM é possível descrever a textura de uma determinada imagem a partir de um conjunto de características. Desta forma, os atributos adquiridos a partir destes métodos se tornam bastante significativos para a extração de características e posterior reconhecimento de falhas em amostras de couro caprino.

A.6.2 Local Binary Patterns (LBP)

Padrões Binários Locais (LBPs) é um descritor de textura que foi popularizado por Ojala, Pietikäinen e Mäenpää (2002). Diferente do que Haralick, Shanmugam e Dinstein (1973) propuseram na GLCM, o LBP calcula uma representação local da textura e não uma representação global, sendo construída comparando cada *pixel* com seus vizinhos.

Na Figura 56, são apresentadas três matrizes, a da esquerda representa parte de uma imagem, uma matriz 2D, com um *pixel* central e vizinhança de 8 *pixels*. A matriz do centro apresenta valores formados a partir da anterior. Nota-se que se a intensidade do *pixel* central for menor ou igual a do vizinho, o *pixel* equivalente na segunda matriz recebe 1, caso contrário, o mesmo será definido como 0. Após o passo anterior, é necessário calcular o valor do LBP, ou seja, transformar os dados da nova matriz em um valor binário e em seguida convertê-lo para decimal. Este novo valor é inserido no elemento correspondente à imagem original, alterando-a.

Figura 56 – Funcionamento do método LBP.



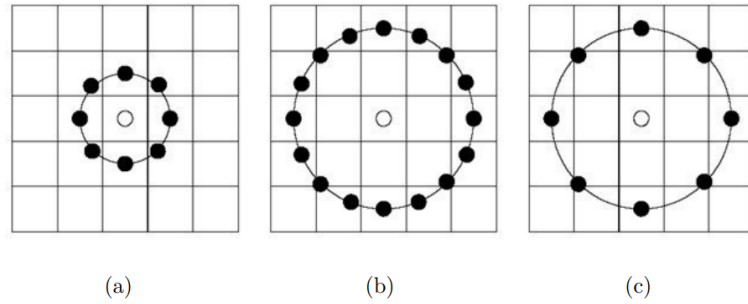
Fonte: Roberto e Souza (2018).

Segundo Rosebrock (2015), um benefício da implementação original do LBP é a possibilidade de capturar detalhes extremamente granulares na imagem. Desta forma, nota-se que é um ótimo extrator para falhas pequenas como picadas de mosquito, rufa, entre outros.

Porém, ser capaz de capturar detalhes em uma escala tão pequena é também uma grande desvantagem do algoritmo, pois não é possível capturar detalhes em escalas variáveis, apenas de forma fixa, ou seja, de acordo com o tamanho escolhido para a máscara.

Para lidar com o problema citado, Ojala, Pietikäinen e Mäenpää (2002) propuseram considerar tamanhos de vizinhança variáveis a partir de dois parâmetros: o número de pontos circularmente simétricos, dessa forma não dependeria mais de uma vizinhança quadrada, e o raio, que permite contabilizar diferentes escalas. Este método pode ser representado como $LBP_{R,P}$ no qual R representa o tamanho do raio e P , o número de pontos. Na Figura 57, são apresentadas algumas variações de valores de raio e números de pontos.

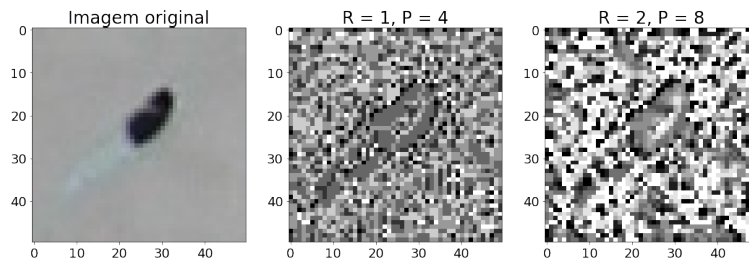
Figura 57 – Exemplos de vizinhanças circulares, (a) $LBP_{1,8}$; (b) $LBP_{2,16}$ e $LBP_{2,8}$.



Fonte: Pietikainen et al. (2011).

Semelhante aos métodos baseados em vizinhos, como os filtros da média e mediana, apresentados anteriormente, o LBP é aplicado a todos os elementos da imagem original, substituindo o valor do elemento da mesma pelo novo valor obtido. Na Figura 58, são apresentados alguns resultados obtidos com variações de tamanho de raio e número de pontos.

Figura 58 – Aplicação do método LBP com diferentes tamanhos de raio e número de pontos.



Fonte: Autor.

Segundo Cavalcante (2016), LBP é um descritor simples e eficiente que rotula os *pixels* de uma imagem, atribuindo um número binário a cada um deles. Estes rótulos compõem um histograma que descreve a textura da imagem ou região em questão. A partir desses dados é possível fazer o treinamento e classificação de imagens.

De acordo com Pietikainen et al. (2011), além da sua simplicidade computacional, uma das propriedades mais importantes do LBP, é a capacidade de invariância, desta forma, caso surjam mudanças nos níveis de cinza, que podem ocorrer devido a mudanças na iluminação do ambiente, a extração de características não se altera. Estes benefícios são fatores chave para a implementação deste método neste trabalho.

ANEXO B – Aprendizagem de Máquina

Configurações complexas de problemas em diferentes aspectos técnicos, comerciais e financeiros evocam uma necessidade crescente de aplicativos de computador que devem mostrar inteligência comportamental (KRUSE et al., 2013).

Burkov (2019) apresenta que o aprendizado de máquina é um subcampo da ciência da computação que se preocupa com a construção de algoritmos que, para serem úteis, contam com uma coleção de exemplos de algum fenômeno. Estes exemplos podem vir da natureza, ações humanas ou gerados por outro algoritmo. O aprendizado de máquina também pode ser definido como o processo de resolver um problema prático por reunir um conjunto de dados e construir algoritmicamente um modelo estatístico com base nesse conjunto de dados. Presume-se que esse modelo estatístico seja usado de alguma forma para resolver o problema prático.

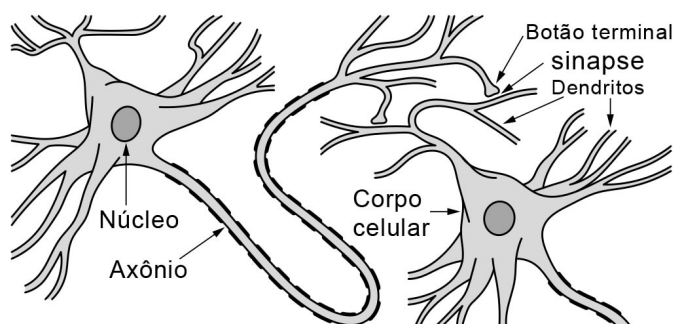
A seguir são apresentados os métodos de aprendizado de máquina utilizados neste documento.

B.1 Perceptron de única camada

Redes neurais artificiais são inspiradas por estruturas e princípios estruturais do sistema nervoso, particularmente, do cérebro de animais e humanos (KRUSE et al., 2013).

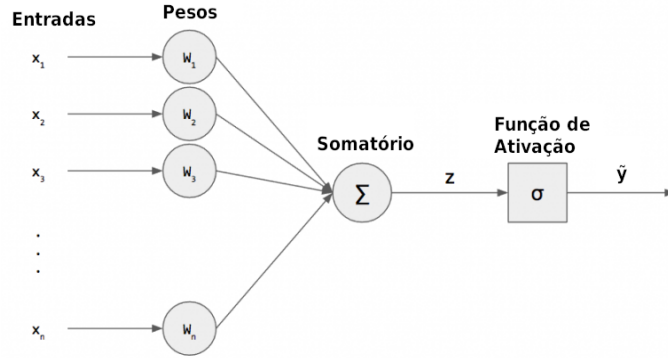
Inspirado no neurônio biológico, apresentado na Figura 59, o neurônio artificial, Figura 60, trata-se de um modelo matemático criado por Rosenblatt (1958). O mesmo é baseado no *threshold logic units*, elaborado por McCulloch e Pitts (1943) e também conhecido como perceptron.

Figura 59 – Estrutura do neurônio biológico.



Fonte: Adaptado de Kruse et al. (2013).

Figura 60 – Estrutura de um neurônio artificial.



Fonte: Adaptado de Deshpande (2017).

Segundo Palmiere (2016), o Perceptron é a forma mais simples da configuração de uma rede neural artificial, uma vez que é constituída de uma única camada neural e de um único neurônio.

Na Equação B.1 é apresentado o funcionamento de uma rede *perceptron*, na qual, $\omega(n)$ representa os pesos sinápticos e $x(n)$ as entradas. Esta fase é conhecida como propagação.

$$\sigma = \text{sinal}[\omega^T(n)x(n)] \quad (\text{B.1})$$

Para a função de ativação, geralmente é utilizada a função sinal, representada na Equação B.2, na qual é responsável por gerar a saída do algoritmo.

$$\text{sinal} = \begin{cases} +1, & \text{se } y(n) > 0 \\ -1, & \text{se } y(n) \leq 0 \end{cases} \quad (\text{B.2})$$

Para calcular o erro ($e(n)$), é utilizada a Equação B.3, na qual $d(n)$ representa a saída desejada e $y(n)$ a saída calculada pelo algoritmo.

$$e(n) = d(n) - y(n) \quad (\text{B.3})$$

Segundo Haykin (2001) "Uma RNA aprende através dos ajustes aplicados a seus pesos sinápticos e níveis de *bias*". Isto posto, uma das formas mais utilizadas é a regra delta, apresentada na Equação B.4, na qual η representa a taxa de aprendizado.

$$\omega(n+1) = \omega(n) + \eta e(n) x(n) \quad (\text{B.4})$$

De acordo com Rosenblatt (1958), a aprendizagem do perceptron converge caso as duas classes possam ser separadas por um hiperplano linear, caso contrário, problemas de convergência serão encontrados pelo classificador.

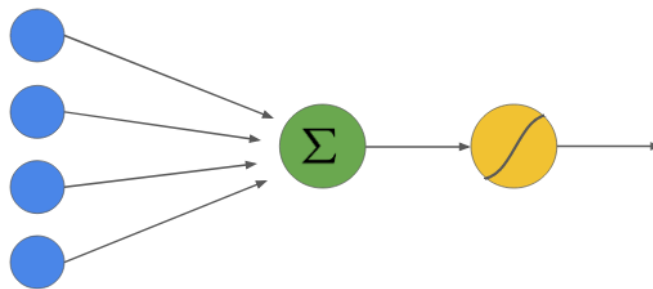
Coppin (2010) explana que os perceptrons podem apenas aprender a modelar funções que sejam linearmente separáveis. Desta forma, em casos nos quais uma função linear não exista, o perceptron não poderá ser bem-sucedido.

Desta forma, fica nítido que a escolha deste algoritmo é apropriada à este trabalho devido a sua simplicidade e para verificação se o problema é linearmente separável.

B.2 Logistic Regression

A técnica *Logistic Regression*, apresentada na Figura 61, está inserida no grupo de algoritmos de aprendizado supervisionado e é basicamente um modelo de classificação probabilística. Este método estatístico foi elaborado com o intuito de prever classes binárias, isto é, uma dicotomia.

Figura 61 – Representação do algoritmo *Logistic Regression*.

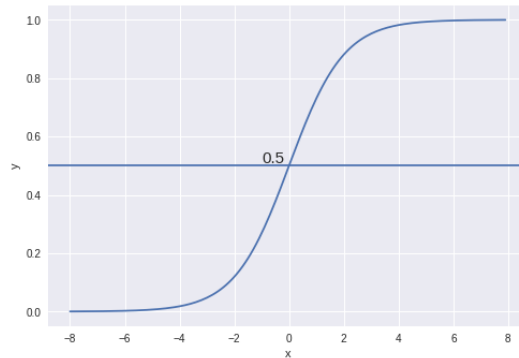


Fonte: Adaptada de Agarwal (2018).

Segundo Raschka (2015), *Logistic Regression* é um modelo de classificação semelhante ao perceptron e pode ser estendido à classificação multiclasse. Neste caso, é denominada como *Multinomial Logistic Regression*.

Conforme pode ser analisado na Figura 62, diferentemente do perceptron, *Logistic Regression* faz uso de uma função sigmóide logística, esta, por sua vez, retorna valores reais entre 0 e 1, que representam o grau de proximidade com a saída correta.

Figura 62 – Função sigmóide logística.



Fonte: Agarwal (2018).

O algoritmo é iniciado a partir da Equação B.5, nos quais os valores w representam os pesos aprendidos do modelo, b o viés e x as entradas de um determinado conjunto de dados.

$$z = b + w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (\text{B.5})$$

Desta forma, para prever a probabilidade de uma determinada amostra pertencer a uma classe específica, utiliza-se a função sigmóide logística, representada na Equação B.6, no qual y representa a saída do modelo de *Logistic Regression*.

$$y = \frac{1}{1 + \exp^{-z}} \quad (\text{B.6})$$

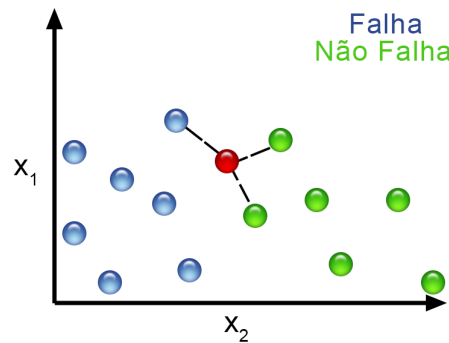
A partir da utilização da função sigmóide, nota-se que os resultados obtidos através desta serão valores reais entre 0 e 1, que representam a probabilidade de um determinado evento ocorrer ou não.

B.3 *K-Nearest Neighbor* (KNN)

Baseado no algoritmo *Nearest Neighbor*, proposto por Cover e Hart (1967), o KNN é um algoritmo de aprendizado baseado em instâncias, ou seja, armazenam dados e usam os mesmos para determinar a classificação de uma entrada.

De acordo com Klein e Bondenseo (2018), o KNN faz parte dos métodos baseados em vizinhos. Este grupo é conhecido como métodos de aprendizado de máquina não generalizantes, pois não há a necessidade de um treinamento.

Para a implementação do KNN, faz-se necessário apenas um conjunto de dados e um parâmetro, denominado K . Desta forma, como pode ser visto na Figura 63, dada uma nova entrada, o algoritmo obtém as classificações dos K vizinhos mais próximos da mesma e atribui a ela, a classificação mais comumente encontrada pelos vizinhos.

Figura 63 – Representação do funcionamento do algoritmo KNN para $K = 3$.

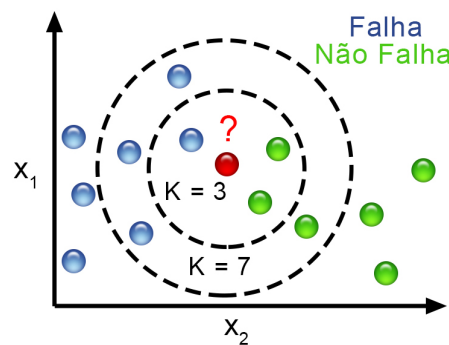
Fonte: Autor.

Durante a implementação do KNN é necessário conhecer o problema ao qual ele será implementado, pois, ao definir o parâmetro K pequeno, a classificação se torna mais precisa, porém, estará mais suscetível a erros, caso o conjunto de dados possua ruídos.

Por outro lado, ao definir um tamanho grande para K , o classificador poderá considerar instâncias relacionadas a outras classes do conjunto de dados.

Ao analisar a Figura 64, percebe-se que ao definir o valor $K = 3$, a nova entrada, caracterizada pela cor vermelha, deverá ser classificada como Não Falha. Porém, ao ajustar o parâmetro com o valor 7, deverá ser definida como Falha. Isto demonstra a importância da escolha correta para o valor do parâmetro.

Figura 64 – Representação do funcionamento do algoritmo KNN.



Fonte: Autor.

Conforme é apresentado nos trabalhos de Amorim et al. (2010), Aquino (2017) e Renato Pereira (2018), o KNN mostra-se um ótimo classificador, possui simplicidade em sua implementação e classifica dados não linearmente separáveis.

Em contrapartida, ao ser dada uma nova entrada, é calculada a sua distância com todos os registros do universo de dados, assim, para um conjunto de dados grande, o classificador tende a exigir um maior tempo computacional.

B.4 Classificador de Fisher baseado na regra de Bayes

O classificador de Fisher (1938) possui um limite de decisão linear. Este é gerado ajustando densidades condicionais de classes aos dados e usando a regra do Teorema de Bayes. O modelo ajusta uma densidade gaussiana a cada classe, assumindo que todas as classes compartilham a mesma matriz de covariância.

Conforme Brownlee (2020), o classificador faz previsões estimando a probabilidade de um novo conjunto de entradas pertencer a cada classe. A classe que obtém a maior probabilidade é a classe de saída e é feita uma previsão. O modelo usa o Teorema de Bayes para estimar as probabilidades.

De acordo com Coppin (2010), o teorema de Bayes pode ser usado para calcular a probabilidade de que um evento ocorra ou que uma proposição seja verdadeira, dado que já é conhecido um fragmento relacionado de informação.

Motwani (2019) cita que o teorema de Bayes faz uso de conhecimento ou experiência prévia para fornecer melhores resultados. Desta forma, o mesmo pode ser declarado conforme a Equação B.7.

$$p(A|B) = \frac{p(A)p(B|A)}{p(B)} \quad (\text{B.7})$$

Beyeler (2017) reescreve o Teorema de Bayes conforme a Equação B.8.

$$posterior = \frac{prior * likelihood}{evidence} \quad (\text{B.8})$$

Nestes termos, considera-se:

- A e B: eventos;
- $p(A/B)$, *posterior*: a probabilidade a ser obtida;
- $p(A)$, *priori*: corresponde ao conhecimento inicial, a probabilidade de que um evento ocorra;
- $p(B/A)$, *likelihood*: a probabilidade condicional de B, dado que A é verdadeiro.
- $p(B)$, *evidence*: a probabilidade do evento B ocorrer.

Considerando X como um conjunto com apenas dois atributos, dado que $X = (x_1, x_2)$, e C como a classe (saída), o teorema pode ser reescrito conforme a Equação B.10.

$$p(C|x_1, x_2) = p(x_1|x_2, C)p(x_2|C)p(C) \quad (\text{B.9})$$

Ao comparar a Equação B.9 com as anteriores, percebe-se a falta do denominador. Isto ocorre, pois, segundo Beyeler (2017), na prática, há interesse apenas no numerador da fração, pois o denominador não depende de C , portanto, é basicamente constante e pode ser negligenciado.

Segundo Fukunaga (1990), o classificador de Bayes pressupõe que vetores de recursos de cada classe sejam normalmente distribuídos (embora não necessariamente distribuídos independentemente). Portanto, supõe-se que toda a função de distribuição de dados seja uma mistura gaussiana, um componente por classe. Usando os dados de treinamento, o algoritmo estima vetores médios e matrizes de covariância para cada classe e, em seguida, os utiliza para previsão.

B.5 Classificador ingênuo de Bayes (*Naïve Bayes*)

Para Coppin (2010), o classificador ingênuo de Bayes é um sistema de aprendizagem simples, mas efetivo no qual cada fragmento de dados a ser classificado consiste em um conjunto de atributos, cada um dos quais podendo assumir diversos valores. Os dados, então, são classificados em uma única categoria.

Segundo Rethana (2018), o *Naïve Bayes* é um algoritmo de classificação usado para classificação binária ou multi-classe. A classificação é realizada calculando as probabilidades posteriores e encontrando a hipótese com maior probabilidade.

Beyeler (2017) cita que problemas do mundo real podem conter diversas variáveis. Desta forma, o cálculo por trás da Equação B.9, citada no classificador de Fisher (Seção B.4), torna-se muito complicado pois cada variável depende de todas as outras.

Desta forma, a ideia do classificador *Naïve Bayes* consiste em assumir que todas as evidências são independentes umas das outras. No entanto, nota-se que tal independência dos atributos raramente ocorre no mundo real. Devido a este fator, o classificador recebe o nome de ingênuo (*Naïve*).

A suposição de que todos os atributos são independentes simplifica a Equação B.9 para B.10.

$$p(C_1|x_1, x_2, \dots, x_n) = p(x_1|C_1)p(x_2|C_1)\dots p(x_n|C_1)p(C_1) \quad (\text{B.10})$$

Conforme apresentado, o classificador *Naïve Bayes* consiste em verificar todas as equações (para todas as classes $C_1, C_2, \dots, C_m$) e depois escolher a classe que possui a maior probabilidade. Este procedimento é conhecido como regra de decisão *maximum a posteriori* (PAM).

Coelho, Richert e Brucher (2018) citam que apesar do nome (*Naïve*), o classificador é, provavelmente um dos algoritmos de aprendizado de máquina mais elegantes que existem para uso prático e prova ser bastante robusto para recursos irrelevantes, que gentilmente ignora, aprendendo rápido e prevendo igualmente, além de não requerer muito armazenamento.

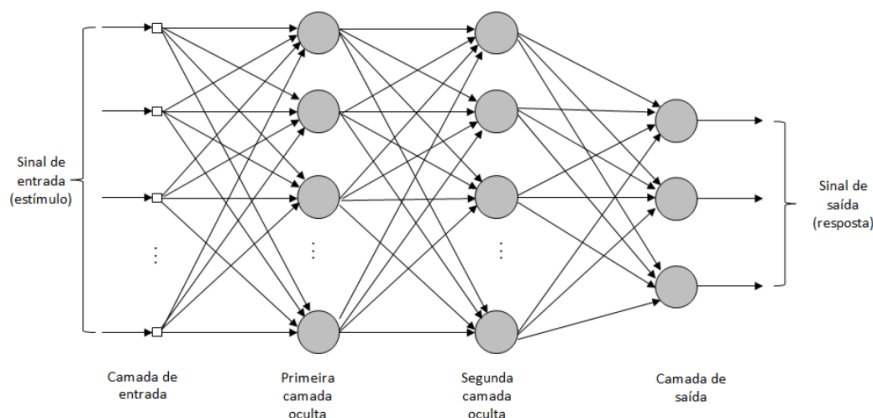
B.6 Multilayer Perceptron (MLP)

O MLP é uma rede neural artificial capaz de solucionar problemas não lineares, fazem parte dos algoritmos de aprendizagem de máquina supervisionados e bioinspirados.

Conforme Backes e Sá (2019), a pesquisa em redes neurais artificiais sofreu um grande decréscimo em virtude de críticas das limitações do perceptron para problemas complexos. Entretanto, Bryson (2018) elaboraram o algoritmo de retropropagação (ou do inglês, *backpropagation*) que permitiu o treinamento de redes perceptron multicamadas e consequentemente o renascimento pelo interesse sobre redes neurais.

Como apresentado na Figura 65, o MLP é uma rede neural composta por mais de um Perceptron e que pode ser dividida em três partes: uma camada de entrada, camadas ocultas e camada de saída.

Figura 65 – Representação de uma Rede Neural Multilayer Perceptron.



Fonte: Haykin (2001).

A camada de entrada é responsável por receber os valores providos dos vetores de características; as camadas ocultas, as quais são consideradas o verdadeiro mecanismo computacional do MLP, são responsáveis pelo processamento não linear do conjunto de dados; e a camada de saída é responsável por gerar um valor ou conjunto de valores que correspondem ao resultado do cálculo da rede.

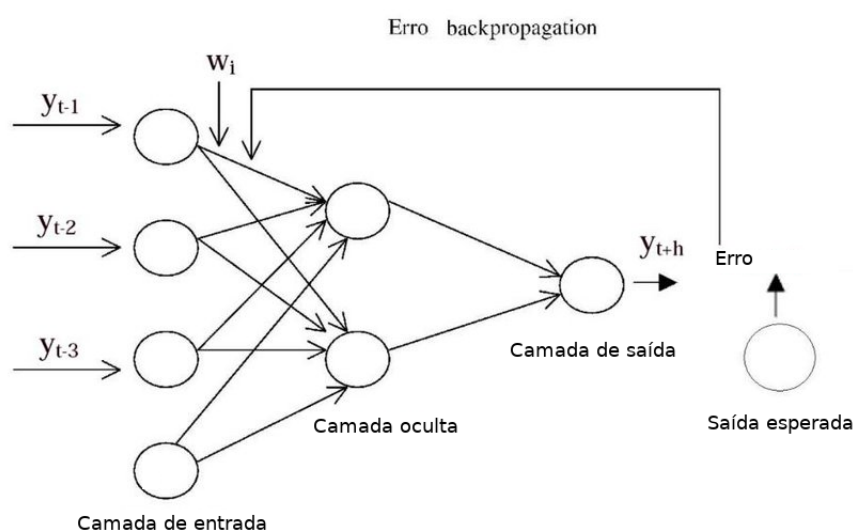
Segundo Coppin (2010), as redes MLP aprendem da mesma forma que os perceptrons simples. A principal diferença é que em uma rede multicamada, cada neurônio possui pesos associados às suas entradas e existe muito mais pesos a serem ajustados quando um erro é cometido para um fragmento de dados de treinamento.

De acordo com Haykin (2001), os perceptrons de múltiplas camadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento de forma supervisionada com um algoritmo muito popular conhecido como algoritmo de retropropagação do erro.

O algoritmo retropropagação do erro é iniciado atribuindo valores aleatórios aos pesos da rede, geralmente pequenos valores na faixa entre -0,5 e 0,5. Desta forma é iniciado o processo de treinamento, no qual envolve a alimentação de dados a partir da camada de entrada até a camada de saída.

Após a obtenção da saída calculada, conforme mostrada na Figura 66, esta, por sua vez, é comparada com a saída correta. Os resultados desta comparação são alimentados de volta, através da rede, fazendo mudanças nos pesos dos nós ao longo do caminho da rede neural. O algoritmo repete esta ação até que as saídas produzidas para os dados de treinamento sejam próximas o suficiente dos valores desejados, ou seja, até que os erros sejam suficientemente pequenos.

Figura 66 – Estrutura de uma RNA do tipo MLP com o algoritmo *backpropagation*.



Fonte: Adaptada de Camelo, Lucio e Junior (2017).

Moreira (2018) afirma que uma das limitações da rede MLP é que, ao aumentar muito o número de camadas e neurônios, ela tende a ficar com um número de parâmetros muito grande e com isso, tão pesada ao ponto de o hardware não conseguir processar e ela não convergir.

B.7 Support Vector Machine (SVM)

Propostas por Vapnik (1995), as máquinas de vetores de suporte são modelos de aprendizado supervisionado com algoritmos de aprendizados associados que analisam os dados usados para classificação e análise de regressão (DATATURKS, 2018).

São amplamente utilizadas devido às suas atrativas características e desempenho de generalização promissora. Algumas destas características são a habilidade em modelar

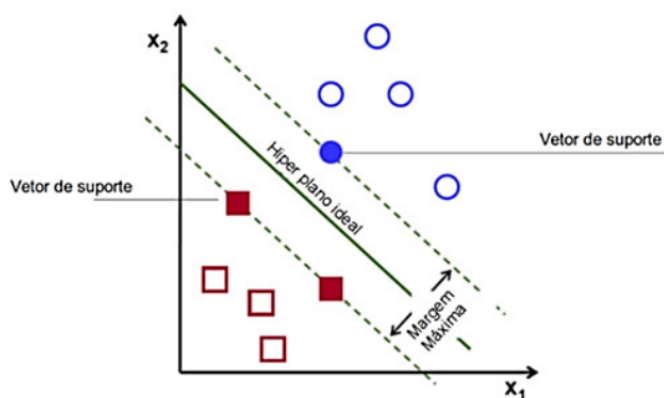
problemas não lineares, pois a sua capacidade de generalização não depende da dimensão do espaço (LIMA, 2014).

Segundo Haykin (2001), a ideia principal de uma máquina de vetor suporte consiste em construir um hiperplano como superfície de decisão, de tal forma que a margem de separação entre as suas classes seja máxima.

Como visto, a ideia principal do SVM é construir um bom hiperplano, de forma que o mesmo possa separar classes diferentes de dados tentando encontrar a maior margem possível. Para que isso ocorra, o classificador recebe como entrada um conjunto de registros já rotulados. Desta forma, as instâncias mais próximas do hiperplano são consideradas como vetores de suporte.

Na Figura 67, é apresentado um conjunto de dados linearmente separável e o hiperplano ideal para o mesmo. Nota-se que, os vetores de suporte, formas preenchidas, conforme Haykin (2001) cita, são os dados mais próximos da superfície de decisão, desta forma, os mais difíceis de classificar. As linhas tracejadas são denominadas como separadores de margem máxima.

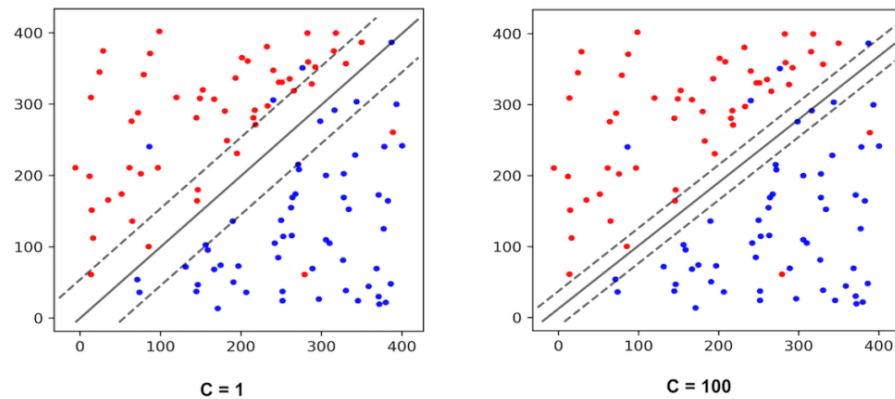
Figura 67 – Exemplo de hiperplano ótimo em um conjunto de dados linearmente separável.



Fonte: Andrade (2015).

O parâmetro C define o limite de decisão do SVM, desta forma, conforme mostrada na Figura 68, nos casos em que o parâmetro C seja pequeno, um hiperplano de margem grande será escolhido e poderá custar um número maior de erros de classificação. Caso contrário, será escolhido um hiperplano de margem menor que tentará classificar muitos outros exemplos corretamente.

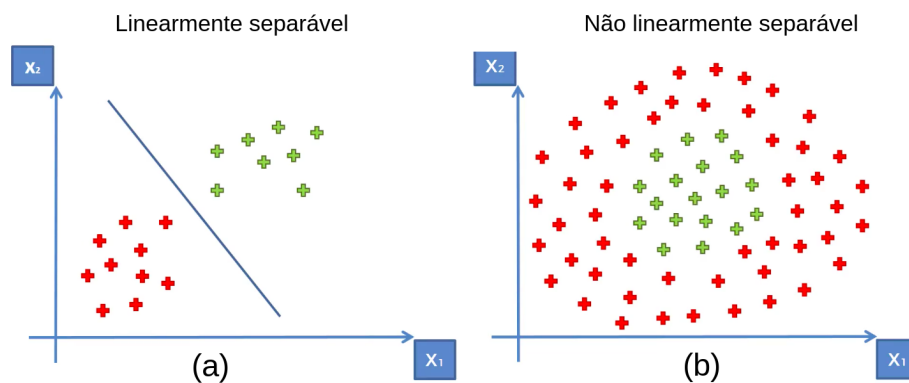
Figura 68 – Exemplo de aplicação do parâmetro C.



Fonte: Shrimali (2018).

O algoritmo de SVM consegue fazer ótimas classificações para problemas linearmente separáveis (Figura 69 (a)), porém, ao analisar problemas do mundo real fica claro que a grande maioria não é linearmente separável (Figura 69 (b)). Visando resolver esta problemática foi desenvolvido o truque de *kernel* (*kernel trick*).

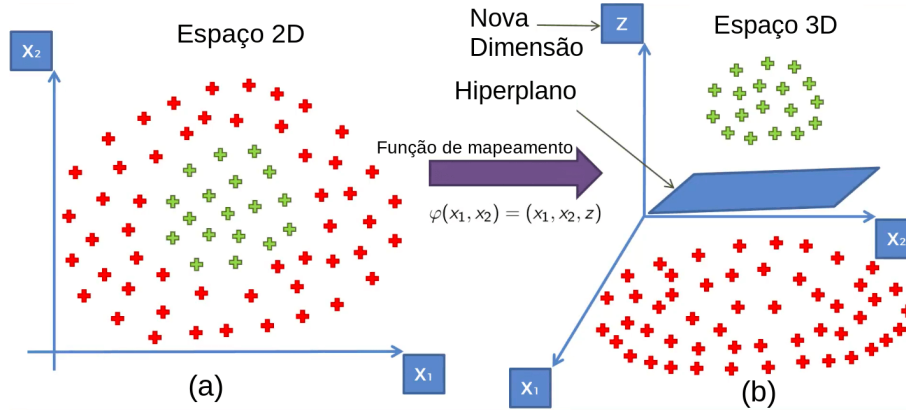
Figura 69 – Exemplo de conjunto de dados linear (a) e não linear(b).



Fonte: Adaptado de Mel (2018).

Para problemas não linearmente separáveis (Figura 70 (a)), o truque de *kernel*, mapeia os dados a partir do espaço original para um espaço de alta dimensionalidade com o intuito de conseguir uma melhor separação entre as classes. (Figura 70 (b)). Desta forma, os métodos de *kernel* são utilizados para encontrar uma superfície de decisão que divida claramente diferentes classes.

Figura 70 – Aplicação do truque de kernel em um conjunto de dados não linear.



Fonte: Adaptado de Mel (2018).

Existem diferentes tipos de *kernels*, entre eles é possível citar, como exemplo; linear, polinomial, gaussiano, entre outros. A seleção do melhor método que se adapte ao conjunto de dados depende muito do problema a ser resolvido.

A seguir, nas Equações B.11, B.12, B.13 e B.14, são apresentados alguns exemplos de *kernel* mais utilizados na literatura.

- Linear:

$$k(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j \quad (\text{B.11})$$

- Polinomial:

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i^T \vec{x}_j + 1)^d \quad d \geq 2 \quad (\text{B.12})$$

- Gaussiano:

$$k(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{2\sigma^2}\right) \quad \sigma > 0 \quad (\text{B.13})$$

- Tangente Hiperbólica (Sigmoide):

$$k(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i^T \vec{x}_j + c) \quad (\text{B.14})$$

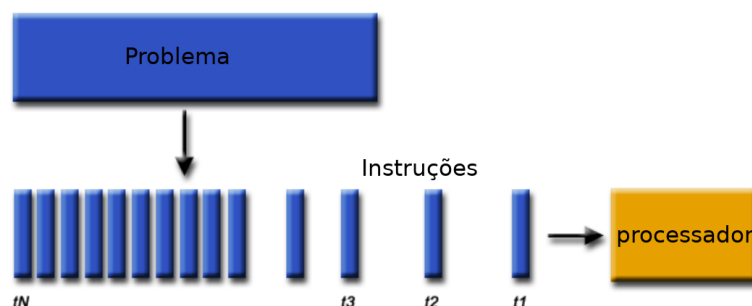
Além das características já citadas, as máquinas de vetores de suporte fazem parte deste trabalho, pois, segundo Yadav (2018), estes algoritmos são bastante eficazes quando o número de recursos for superior a exemplos de treinamento. São adequados para classificação binária e o hiperplano é afetado apenas pelos vetores de suporte, portanto, valores discrepantes têm menos impacto.

ANEXO C – Computação Paralela

De acordo com Pierfederici (2016), definições de computação paralela são abundantes. No entanto, em uma definição simples, pode-se afirmar que é o uso simultâneo de mais de um processador para resolver um problema. Essa definição é mais especializada, exigindo que os processadores residam na mesma placa mãe. Isto ocorre com o intuito, principalmente, de distinguir computação paralela de computação distribuída.

Tradicionalmente, um *software* é escrito utilizando computação serial, assim, como apresentado na Figura 71, um determinado problema é dividido em várias instruções nas quais são executadas de forma sequencial e utilizando apenas um processador. Desta forma, apenas uma instrução pode ser executada a cada momento.

Figura 71 – Problema sendo solucionado de forma sequencial.



Fonte: Adaptada de Barney (2019).

Conforme Haff (2011), o problema é que um programa projetado para executar um único encadeamento de instruções não fica mais rápido, caso apenas adicione *hardware* para fazer mais trabalho de forma independente. Para reduzir o tempo que o programa leva para realizar uma determinada tarefa, ele deve ser dividido em partes nas quais o processador possa trabalhar paralelamente.

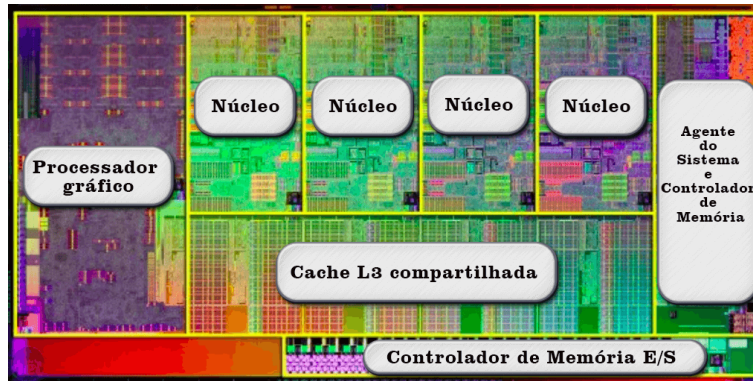
Analogamente, Lanaro (2017) cita que com o processamento paralelo, é possível aumentar a quantidade de cálculos que o programa pode fazer em um determinado momento sem precisar de um processador mais rápido. A ideia principal é dividir uma tarefa em várias subunidades e empregar vários processadores para resolvê-las independentemente.

Barney (2019) cita que hoje praticamente todos os computadores independentes são paralelos na perspectiva do hardware. Desta forma, os mesmos possuem várias unidades funcionais como, por exemplo, múltiplas unidades/núcleos de execução, cache, processamento gráfico (GPU), etc.

Na Figura 72, é apresentado um processador *Quad-core Intel Sandy Bridge*, o mesmo é produzido desde 2011. Como citado, possui vários núcleos, o que permite a

possibilidade de processar algoritmos paralelos.

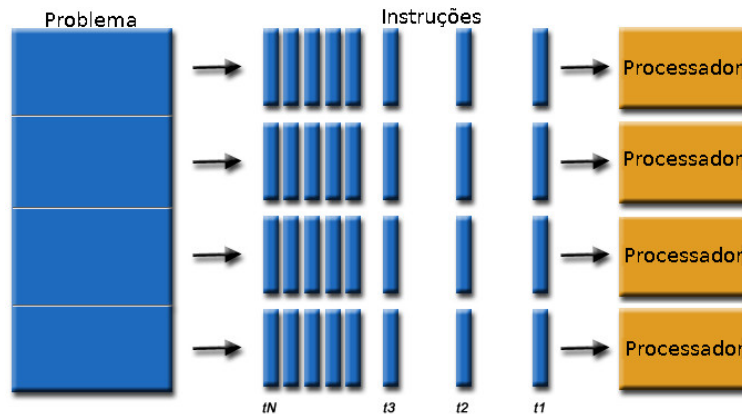
Figura 72 – Processador *Quad-core Intel Sandy Bridge*.



Fonte: Adaptada de Haff (2011).

Deste modo, na Figura 73 é mostrada, no sentido mais simples, a ideia de computação paralela. Nota-se que um determinado problema é dividido em várias partes discretas. Cada parte é dividida em diversas instruções para, posteriormente, serem executadas simultaneamente por diferentes processadores.

Figura 73 – Problema sendo solucionado de forma paralela.



Fonte: Adaptada de Barney (2019).

Diversas são as métricas utilizadas para mensurar o desempenho de um algoritmo paralelo, alguns exemplos são:

- Tempo de execução;
- Aceleração: indica quantas vezes o programa paralelo é mais rápido que a versão sequencial, ou seja, é a aceleração observada de um código paralelo.

A sua formulação, tal qual mostrada na Equação C.1, se dá através do tempo gasto para o processamento em uma única máquina/processador ($T(1)$) e o tempo da execução paralela ($T(p)$), ou seja, utilizando p máquinas/processadores.

$$S(p) = T(1)/T(p) \quad (C.1)$$

-
- Eficiência: conforme apresentada na Equação C.2, é a taxa de utilização média das unidades de processamento. É a relação entre o *Speedup* observado e o número de processadores.

$$E(p) = S(p)/p = T(1)/(p * T(p)) \quad (C.2)$$

- Escalabilidade: É a análise do tempo de execução e da eficiência e de como os mesmos variam com o aumento da quantidade de processadores.

Como é possível notar, comparada à computação serial, a computação paralela é bastante adequada para modelar e simular fenômenos complexos do mundo real.

Computadores modernos são paralelos em sua arquitetura, ou seja, possuem vários processadores/núcleos e na maioria dos casos, programas seriais são executados em computadores atuais, o que acarreta no desperdício do potencial da computação.

ANEXO D – Detecção de Áreas Livres de Falhas

A detecção das áreas livres de falhas baseia-se em um problema de programação dinâmica explanada por Dhaval Dave (2017), a qual localiza, em uma matriz binária, a maior submatriz quadrada de 1's existente na mesma. Este procedimento é descrito a seguir.

Considerando uma matriz (Figura 74), na qual 0's representam falhas em peças de couro e 1's, regiões sem falha, a detecção de áreas livres de falhas é iniciada a partir da localização do maior agrupamento de valores 1's nesta matriz.

Figura 74 – Exemplo de uma imagem digital contendo defeitos (0) e regiões boas (1).

0	1	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
0	1	1	1	0	0
1	1	1	1	1	0
1	0	1	0	1	1
1	1	0	0	1	0

Fonte: Autor.

O primeiro passo é elaborar uma matriz auxiliar (MA) e nela inserir os dados contidos na primeira linha e coluna da matriz original (MO), conforme ilustrado na Figura 75. Nesta matriz, o par (i, j) representa os índices da mesma.

Figura 75 – Primeira linha e coluna de uma determinada imagem sendo copiada para uma matriz auxiliar.

0	1	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
0	1	1	1	0	0
1	1	1	1	1	0
1	0	1	0	1	1
1	1	0	0	1	0

→

0	1	1	0	1	0
0					
0					
0					
1					
1					
1					

Fonte: Autor.

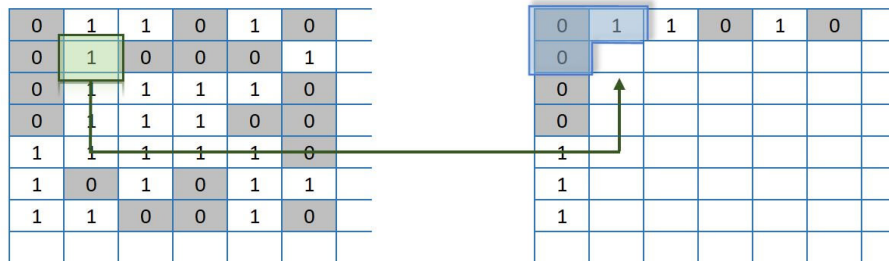
Neste momento é iniciado o processo de verificação dos *pixels*. A cada valor 0 encontrado na matriz original, o mesmo será repetido na respectiva posição na matriz auxiliar. Em casos que o valor 1 seja encontrado, é realizado um cálculo, conforme

apresentado na Equação D.1, e inserido seu resultado na localização respectiva na matriz auxiliar.

$$MA(i, j) = \min(MA(i - 1, j), MA(i - 1, j - 1), MA(i, j - 1)) + 1 \quad (D.1)$$

O funcionamento da Equação D.1 consiste em localizar o valor mínimo entre as posições $MA(i - 1, j)$, $MA(i - 1, j - 1)$ e $MA(i, j - 1)$ somado ao valor 1 e inserido na posição $MA(i, j)$. Na Figura 76 é apresentado um exemplo deste processo.

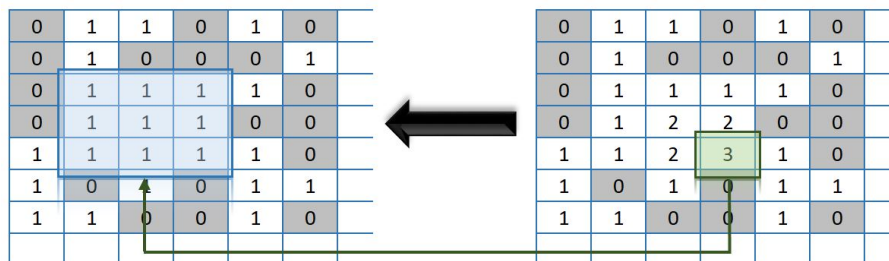
Figura 76 – Representação pictográfica de regiões da pele de caprinos.



Fonte: Autor.

Conforme ilustrado na Figura 77, o maior número obtido a partir do método citado possui valor 3, que representa uma área quadrada de 3×3 .

Figura 77 – Representação pictográfica de regiões da pele de caprinos.



Fonte: Autor.

Dessa forma, a partir do maior valor encontrado é possível obter a maior área quadrada que não possui falhas. Para a resolução do problema de detecção de áreas livres de falhas é elaborada uma função para o programa computacional, esta é apresentada no Algoritmo 1.

Algoritmo 1 Detecção da maior região isenta de falhas

```

1: função DETECTARREGIAO(MO, MA)
2:   para  $i \leftarrow 1$  até  $N$  faça                                 $\triangleright N = \text{Número de linhas}$ 
3:     para  $j \leftarrow 1$  até  $M$  faça                                 $\triangleright M = \text{Número de colunas}$ 
4:       se  $MO(i, j) = 0$  então
5:          $MA(i, j) \leftarrow 0$ 
6:       senão
7:          $MA(i, j) \leftarrow \min(MA(i - 1, j), MA(i - 1, j - 1), MA(i, j - 1)) + 1$ 
8:       fim se
9:     fim para
10:  fim para
11:  devolve  $MA$ 
12: fim função

```

Como explanado, toda a matriz original (MO) é percorrida, analisado os valores contidos nela e, se necessário, efetuado o cálculo apresentado na Equação D.1 de forma a preencher toda a matriz auxiliar (MA). Este processo pode ser executado repetidas vezes com o intuito de obter diversas áreas isentas de falhas.