

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

Fausto Sampaio

**Estudo de redução do custo computacional em aplicações
embarcadas utilizando Máquinas de Vetores Suporte e extratores de
atributos de imagens por textura**

Fortaleza – CE

5 de fevereiro de 2018

Fausto Sampaio

Estudo de redução do custo computacional em aplicações embarcadas utilizando Máquinas de Vetores Suporte e extratores de atributos de imagens por textura

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: Engenharia de Software

Orientador: Prof. Dr. Elias Teodoro da Silva Júnior

Coorientador: Prof. Dr. Pedro Pedrosa Rebouças Filho

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

5 de fevereiro de 2018

Dados Internacionais de Catalogação na Publicação

Instituto Federal do Ceará - IFCE

Sistema de Bibliotecas - SIBI

Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

S192e Sampaio, Fausto.

Estudo de redução do custo computacional em aplicações embarcadas utilizando Máquinas de Vetores Suporte e extratores de atributos de imagens por textura / Fausto Sampaio. - 2017.

100 f.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus Fortaleza, 2017.

Orientação: Prof. Dr. Elias Teodoro da Silva Júnior.

Coorientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.

1. Sistemas embarcados. 2. Visão Computacional. 3. Reconhecimento de Padrões. 4. Avaliação de Desempenho. 5. SVM, GLCM e LBP. I. Título.

CDD 004



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)

Fausto Sampaio

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

**Orientador: Prof. Dr. Elias Teodoro da Silva
Júnior**

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

**Coorientador: Prof. Dr. Pedro Pedrosa
Rebouças Filho**

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

Prof. Dr. Ajalmar Rêgo da Rocha Neto

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

Prof. Dr. Marco Aurélio Wehrmeister

Universidade Tecnológica Federal do Paraná - UTFPR

Fortaleza – CE
5 de fevereiro de 2018

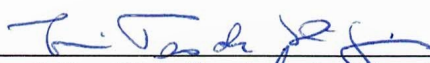
FAUSTO SAMPAIO

Estudo de redução do custo computacional em aplicações embarcadas utilizando Máquinas de Vetores Suporte e extratores de atributos de imagens por textura

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.

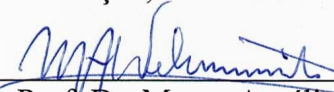
Aprovada em 14 / 12 / 2017.

BANCA EXAMINADORA


Prof. Dr. Elias Teodoro da Silva Júnior (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - IFCE


Prof. Dr. Pedro Pedrosa Rebouças Filho (Co-orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. Ajalmar Rêgo da Rocha Neto
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. Marco Aurélio Wehrmeister
Universidade Tecnológica Federal do Paraná - UTFPR

*Este trabalho é dedicado à minha família,
em especial meus pais José Renato Raulino (in memoriam) e Maria da Conceição
Sampaio, irmãos Ana Flávia Sampaio, Leandro Sampaio e Renata Sampaio Raulino e
minha esposa Andreliny Barros Studart Sampaio.*

Agradecimentos

Quero deixar aqui registrado a minha mais sincera gratidão para com todos os que colaboraram, diretamente ou indiretamente, neste trabalho:

A Deus, que me orienta em todos os momentos.

Aos professores Dr. Elias T. Silva Jr. e Dr. Pedro Pedrosa Rebouças Filho, pelos ensinamentos, orientações, amizade, confiança, paciência e dedicação depositados ao longo deste trabalho.

A banca composta pelos professores Dr. Ajalmar Rêgo da Rocha Neto (IFCE) e Dr. Marco Aurélio Wehrmeister (UTFPR) pelas contribuições oferecidas ao trabalho.

Aos meus pais José Renato Raulino (*in memoriam*) e Maria da Conceição Sampaio, irmãs Renata Sampaio Raulino e Ana Flávia Sampaio, irmãos Leandro Sampaio e Flávio Sampaio pelo apoio incondicional, conselhos, força, amor e amizade sem igual durante todos esses anos.

À minha esposa, Andreliny Barros Studart Sampaio, pelo amor, carinho, compreensão e incentivo para que eu pudesse concluir esta importante etapa da minha vida.

Aos meus amigos Leandro B. Marinho, Kauê Menezes, Leonildo Ferreira, Renata Pereira, Flávio Renato e Juciaris Medeiros pelo apoio e amizade antes e durante os anos de graduação e pós-graduação.

Ao Lucas Costa da Silva pela ajuda nos experimentos da pesquisa.

A todos docentes do Programa de Pós-Graduação em Ciência da Computação (PPGCC-IFCE) e da Graduação em Ciência da Computação (IFCE Campus Maracanaú), pelo ensino de boa qualidade, atenção e apoio.

Ao Prof. Dr. Sandro C. S. Jucá, pelo incentivo, apoio e orientação no curso de graduação.

A todos os amigos da Secretaria de Educação de Maracanaú, de modo especial ao diretor executivo Antônio Nilson Gomes Moreira e chefe do setor de TI João Batista Monteiro ao me liberarem para cursar o mestrado em tempo integral.

“Lembre-se que as pessoas podem tirar tudo de você, menos o seu conhecimento.”
(Albert Einstein)

Resumo

Aplicações embarcadas de visão computacional têm sido incorporadas largamente na robótica e na automação industrial, com o objetivo de melhorar a qualidade e segurança dos processos. Tais sistemas envolvem classificadores de padrões para funções específicas, que muitas vezes apresentam elevado custo de memória e tempo de processamento. Este trabalho apresenta uma análise das restrições de embarcar um classificador que combina máquina de vetores de suporte (*Support Vector Machine*, SVM) com dois extratores de atributos. Um que usa a técnica de extração de características de imagens digitais a partir de uma matriz de coocorrência de níveis de cinza (*Gray Level Co-Occurrence Matrix*, GLCM) e o outro baseado em padrões binários locais (*Local Binary Patterns*, LBP). São investigadas as relações entre os atributos extraídos da imagem e o consumo de recursos em um computador embarcado, sempre com a finalidade de preservar a taxa de acerto do classificador. Como estudos de caso são usadas duas aplicações, classificação da qualidade de couro caprino e localização e navegação de um robô móvel. A metodologia proposta neste trabalho apresenta um estudo completo de viabilidade para embarcar aplicações utilizando SVM em conjunto com os métodos GLCM e LBP em aplicações diversas a partir de avaliações de memória e desempenho. Para reduzir os custos computacionais foram avaliados fatores como tamanho da imagem e quantidade de atributos, além de outros fatores específicos de cada método de extração de atributos. Na classificação foram feitos experimentos para avaliar o tempo de computação, consumo de memória e desempenho do classificador. Para a classificação das imagens de couro caprino, os tempos de computação obtidos foram de 3,1s e 0,54s, e consumos de memória de 16,039 kB e 0,010 kB com acurácias superiores a 90% na classificação de imagens com resolução 47×63 pixels com o uso de GLCM e LBP, respectivamente. Já para a localização do robô móvel foram obtidos tempos de computação de 3,2s e 0,45s, e consumos de memória de 1,039 kB e 0,010 kB com acurácias próximas a 100% utilizando imagens de 40×30 pixels. Esses resultados permitem utilizar plataformas baseadas em microcontroladores para aplicações de visão computacional, reduzindo custo de desenvolvimento e de implantação.

Palavras-chave: Sistemas embarcados. Visão Computacional, Reconhecimento de Padrões, SVM, GLCM. LBP. Avaliação de Desempenho.

Abstract

Embedded applications of computer vision have been incorporated largely in robotics and industrial automation, with the aim of improving the quality and safety of the processes. Such systems involve pattern classifiers for specific functions, which often have high memory costs and processing time. This work presents an analysis of the constraints on embedding a classifier that combines support vector machine (SVM) with two feature extractors. One that uses the technique of extracting characteristics of digital images from a Gray Level Co-Occurrence Matrix (GLCM) and the other based on Local Binary Patterns (LBP). The relationships between the features extracted from the image and the resources consumption in an embedded computer are investigated, always with the purpose of preserving the classifier hit rate. As case studies two applications are used, quality classification of goat leather and location and navigation of a mobile robot. The methodology proposed in this work presents a complete feasibility study to embed applications using SVM in conjunction with the GLCM and LBP methods in diverse applications from memory and performance evaluations. In order to reduce computational costs, factors such as image size and number of features were evaluated, as well as other factors specific to each method of feature extraction. In the classification experiments were made in order to evaluate computing time, memory consumption and classifier performance. For the classification of the goat leather images, the computing times were 3.1s and 0.54s, and memory consumptions of 16.039 kB and 0.010 kB with accuracy greater than 90% in the classification of images with resolution 47×63 pixels using GLCM and LBP, respectively. As for the location of the mobile robot, computing times of 3.2s and 0.45s were obtained, and memory consumptions of 1.039 kB and 0.010 kB with approximations of 100% using images of 40×30 pixels. These results allow the use of microcontroller based platforms for computer vision applications, reducing development and deployment costs.

Keywords: embedded systems. computer vision. pattern recognition. SVM. GLCM. LBP. performance evaluation.

Lista de ilustrações

Figura 1 – Imagem representada como uma matriz de intensidade visual (GONZALEZ; WOODS, 2010).	30
Figura 2 – Passos fundamentais em processamento de imagens digitais (GONZALEZ; WOODS, 2010).	30
Figura 3 – Etapas de um sistema de visão computacional (MARQUES FILHO; VIEIRA NETO, 1999).	31
Figura 4 – Representação da GLCM	32
Figura 5 – Exemplo do cálculo do LBP. Para uma vizinhança 3×3 na imagem, o padrão binário local é determinado pela comparação do <i>pixel</i> central com seus vizinhos na ordem fixada atribuída pela seta. O padrão binário gera um número que é computado no histograma LBP.	36
Figura 6 – Indução de classificador em aprendizado supervisionado (LORENA; CARVALHO, 2007).	39
Figura 7 – Exemplos de padrões linearmente e não-linearmente separável respectivamente (GONCALVES, 2015).	40
Figura 8 – Fluxo dos procedimentos adotados neste trabalho.	49
Figura 9 – Exemplos de amostras de imagens das classes do tipo 1 e 7 usadas na aplicação AP1.	51
Figura 10 – Mapa topológico do ambiente de navegação (MARINHO, 2017).	52
Figura 11 – Exemplos de amostras de imagens das classe do tipo 1 e 15 usadas na aplicação AP2.	52
Figura 12 – Exemplo do processo de redimensionamento de imagem por decimação com $M = 3$	53
Figura 13 – Exemplos de imagens redimensionada. Da esquerda para a direita as imagens são redimensionadas com $M = \{1, 35, 100\}$. A primeira linha corresponde ao processo com uma imagem da classe 7 presente na base de dados BD1. A segunda linha utiliza uma imagem da classe 7 presente na base de dados BD2.	54
Figura 14 – Processo de extração da área do couro.	55
Figura 15 – Ilustração da utilização do algoritmo CFS.	57
Figura 16 – Exemplo do cálculo do LBP considerando 4 <i>pixels</i> vizinhos.	60
Figura 17 – Ilustração do redimensionamento da matriz de coocorrência (CM)	62
Figura 18 – Desempenho do classificador SVM com o extrator GLCM na aplicação AP1, utilizando imagens redimensionadas.	71
Figura 19 – Desempenho do classificador com GLCM para a aplicação AP2 utilizando imagens redimensionadas.	73

Figura 20 – Desempenho do classificador com LBP para a aplicação AP1 utilizando imagens redimensionadas.	75
Figura 21 – Desempenho do classificador com LBP para a aplicação AP2 utilizando imagens redimensionadas.	77
Figura 22 – Desempenho do classificador com a GLCM redimensionada.	82
Figura 23 – Histograma de imagens das aplicações AP1 e AP2, respectivamente. . .	83
Figura 24 – Sistema de Reconhecimento de Padrões usado no trabalho em uma plataforma embarcada.	84
Figura 25 – Tempos de computação (em segundos) na plataforma embarcada para a aplicação AP1.	86
Figura 26 – Tempos de computação na plataforma embarcada para a aplicação AP2.	88

Lista de tabelas

Tabela 1	– Resumo dos <i>kernels</i> mais populares (ROCHA NETO, 2011).	40
Tabela 2	– Resultado classificação imagens Wet Blue (AMORIM, 2009).	45
Tabela 3	– Resultados de SANTOS FILHO (2013) para desempenho dos classificadores com diversos métodos de extração de características (Imagens tamanho 100x100).	46
Tabela 4	– Relação das 24 medidas estatísticas da GLCM.	56
Tabela 5	– Conjuntos de características resultantes do algoritmo CFS por aplicação.	58
Tabela 6	– Relação dos atributos agrupados por custo de complexidade temporal.	59
Tabela 7	– Relação dos atributos agrupados por custo de complexidade espacial.	59
Tabela 8	– Conjuntos de atributos utilizados na dissertação.	61
Tabela 9	– Exemplo de uma matriz de confusão para um problema de 7 classes analisando a classe 1.	64
Tabela 10	– Exemplo de uma matriz de confusão para um problema de 7 classes analisando a classe 4.	64
Tabela 11	– Exemplo de uma matriz de confusão com valores reais.	66
Tabela 12	– Tempos de computação (em segundos) na estação de trabalho para as aplicações AP1 e AP2. Utilização do extrator GLCM com imagens no tamanho original.	68
Tabela 13	– Tempos de computação (em segundos) na estação de trabalho para as aplicações AP1 e AP2. Utilização do extrator LBP com imagens no tamanho original.	69
Tabela 14	– Tempos de computação (em segundos) das melhores soluções GLCM e LBP na estação de trabalho para as aplicações AP1 e AP2. Utilização das imagens no tamanho original.	70
Tabela 15	– Desempenho do classificador SVM combinado com o extrator GLCM usando o conjunto de atributos A3 na aplicação AP1.	72
Tabela 16	– Desempenho do classificador SVM por classe, utilizando o conjunto de atributos A3 e imagens redimensionadas por $M = 73$ (47×63) <i>pixels</i> .	72
Tabela 17	– Desempenho do classificador SVM combinado com o extrator GLCM usando o conjunto de atributos C3 na aplicação AP2.	74
Tabela 18	– Desempenho do classificador SVM por classe, utilizando o conjunto de atributos C3 e imagens redimensionadas por $M = 38$ (105×78) <i>pixels</i> .	74
Tabela 19	– Desempenho do classificador SVM combinado com o extrator LBP usando o conjunto de atributos B4 na aplicação AP1.	76
Tabela 20	– Desempenho do classificador SVM por classe, utilizando o conjunto de atributos B4 e imagens redimensionadas por $M = 10$ (345×460) <i>pixels</i> .	76

Tabela 21 – Desempenho do classificador SVM combinado com o extrator LBP usando o conjunto de atributos D4 na aplicação AP2.	77
Tabela 22 – Desempenho do classificador SVM por classe, utilizando o conjunto de atributo D4 e imagens redimensionas por $M = 10$ (40×30) <i>pixels</i> . . .	78
Tabela 23 – Conjuntos de atributos e tamanhos das imagens considerados ideais para as aplicações.	79
Tabela 24 – Consumos estimativos de memória (em <i>kilobytes</i>) das técnicas GLCM, LBP e SVM na aplicação AP1 utilizando imagens redimensionadas. V representa o total de vetores de suporte.	80
Tabela 25 – Consumos estimativos de memória (em <i>kilobytes</i>) das técnicas GLCM, LBP e SVM na aplicação AP2 utilizando imagens redimensionadas. V representa o total de vetores de suporte.	81
Tabela 26 – Tempo de computação (em segundos) por etapas na aplicação AP1. . .	87
Tabela 27 – Tempo de computação (em segundos) por etapas na aplicação AP2. . .	89

Lista de abreviaturas e siglas

Ac	Acurácia
AM	Aprendizagem de Máquina
AP1	Aplicação 1 (Classificação de Couro Caprino)
AP2	Aplicação 2 (Localização e Navegação de um Robô Móvel)
ASM	Angular Second Moment (Segundo Momento Angular)
BCM	Base de Controle Móvel
BD1	Base de Dados da Aplicação AP1
BD2	Base de Dados da Aplicação AP2
CEM	Consumo Estimativo de Memória
CFS	Correlation Based Feature Selection (Seleção de Atributos Baseada em Correlação)
DSE	Design Space Exploration (Exploração do Espaço de Projeto)
ELM	Extreme Learning Machine (Máquina de Aprendizado Extremo)
Es	Especificidade
ET1	Estação de Trabalho 1 (Matlab)
ET2	Estação de Trabalho 2 (OpenCV)
FN	Falso Negativo
FP	Falso Positivo
FPGA	Field Programmable Gate Array (Arranjo de Portas Programáveis em Campo)
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
IBGE	Instituto Brasileiro de Geografia e Estatística
IMC	Information Measures of Correlation (Medidas de informação da correlação)

GLCM	Gray-Level Co-occurrence Matrix (Matriz de Coocorrência de Níveis de Cinza)
HOG	Histogram of Oriented Gradients (Histograma de Gradientes Orientados)
LBP	Local Binary Pattern (Padrão Binário Local)
MC	Matriz de Coocorrência
MCC	Maximal Correlation Coefficient (Coeficiente de Correlação Máximo)
MLM	Minimal Learning Machine (Máquina de Aprendizado Mínimo)
MLP	Multilayer Perceptron (Perceptron de Múltiplas Camadas)
NMF	Non-negative Matrix Factorization (Fatoração em Matrizes Não-Negativas)
PCA	Principal Component Analysis (Análise de Componentes Principais)
PDI	Processamento Digital de Imagens
RBF	Radial Basis Function (Função de Base Radial)
RLGC	Regularized Linear Gaussian Classifier (Classificador Gaussiano Linear Regularizado)
RNA	Redes Neurais Artificiais
ROI	Region of Interest (Região de Interesse)
Se	Sensibilidade
SMO	Sequential Minimal Optimization (Otimização Mínima Sequencial)
SRP	Sistemas de Reconhecimento de Padrões
SVM	Support Vector Machines (Máquinas de Vetores de Suporte)
TAE	Teoria de Aprendizado Estatístico
TCT	Todos-Contra-Todos
UCT	Um-Contra-Todos
VECM	Valor Estimativo de Consumo de Memória
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
VPP	Valor Preditivo Positivo

Lista de símbolos

$f(x, y)$	Função bidimensional de intensidade de luz (imagem)
D	Distância para o par de níveis de cinza i e j
θ	Direção para o par de níveis de cinza i e j
R	Constante padronizada
P	Número de <i>pixels</i> vizinhos (LBP)
k	rótulo de uma classe
N_g	Número de níveis de cinza
$I(i, j)$	Matriz bidimensional de intensidade de luz (imagem)
σ	Constante para a SVM
$K(., .)$	Função kernel
exp	Função exponencial natural
$s(.)$	Função degrau
max	Função máximo
x	Vetor de atributos
$(\mathbf{x}, y)$	Padrão ou amostra
y	Saída do classificador
M	Número de amostras na decimação
Θ	Notação assintótica para descrever complexidade dos algoritmos
F	Fator de redução da GLCM
L	Número de linhas da imagem
C	Número de colunas da imagem

Sumário

1	INTRODUÇÃO	25
1.1	Motivação	26
1.2	Hipóteses de Trabalho	27
1.3	Objetivos Geral e Específicos	27
1.4	Relevância	28
1.5	Problematização	28
1.6	Estrutura da Dissertação	28
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	Processamento Digital de Imagens - PDI	29
2.2	Extratores de Características	31
2.2.1	Matriz de Coocorrência de Níveis de Cinza	32
2.2.2	Padrões Binários Locais	35
2.3	Reconhecimento de Padrões	37
2.4	Máquina de Vetores Suporte	39
2.4.1	Classificação Multiclasse	41
2.4.1.1	Abordagem Um-Contra-Todos	41
2.4.1.2	Abordagem Todos-Contra-Todos	41
2.5	Projeto de Sistemas Embarcados	42
3	TRABALHOS RELACIONADOS	43
3.1	Estado da Arte no processo de embarcar o classificador SVM	43
3.2	Extratores de atributos GLCM e LBP Embarcados	44
3.3	Estado da Arte na classificação de peças de couro	45
3.4	Estado da Arte na localização e navegação de robôs móveis	46
3.5	Considerações	47
4	METODOLOGIA	49
4.1	Aplicações	50
4.1.1	Classificação automática de couro caprino – AP1	50
4.1.1.1	Base de Dados - BD1	50
4.1.2	Localização e navegação de robôs móveis em mapas topológicos utilizando imagens convencionais – AP2	51
4.1.2.1	Base de Dados - BD2	52
4.2	Redimensionamento das Imagens	53
4.3	Segmentação das Imagens	54

4.4	Extração, seleção e redução de atributos	55
4.4.1	Seleção de atributos baseada em correlação (CFS)	57
4.4.2	Estudo da complexidade na extração dos atributos	58
4.5	Redução da Matriz de Coocorrência (MC)	62
4.6	Classificação e Avaliação	62
4.6.1	Treinamento	63
4.6.2	Testes	63
4.6.2.1	Métricas de Avaliação	63
4.7	Conclusão	66
5	RESULTADOS E DISCUSSÕES	67
5.1	Desempenho do Classificador nas Estações de Trabalho	67
5.1.1	Custo de Computação	68
5.1.1.1	GLCM	68
5.1.1.2	LBP	69
5.1.1.3	Discussão parcial dos resultados	69
5.1.2	Resolução das imagens	70
5.1.2.1	GLCM	70
5.1.2.2	LBP	74
5.1.2.3	Discussão parcial dos resultados	78
5.1.3	Escolha do tamanho ideal da imagem e do conjunto de atributos	78
5.1.4	Consumo de memória	79
5.1.4.1	Estudo analítico	79
5.1.4.2	Redução da Matriz de Coocorrência (MC)	82
5.2	Resultados na plataforma embarcada	83
5.2.1	Validação de resultados	85
5.2.2	Medições de tempo de computação	85
5.2.3	Resultados na aplicação AP1	86
5.2.4	Resultados na aplicação AP2	87
5.3	Análise Geral dos Resultados	89
6	CONCLUSÃO	91
6.0.1	Proposta de trabalhos futuros	92
	REFERÊNCIAS	93

1 Introdução

Um sistema embarcado é qualquer dispositivo que inclui um computador programável, mas não se destina a ser um computador de propósito geral. Os sistemas computacionais embarcados estão presentes em praticamente todas as atividades humanas (WOLF, 2016). Exemplos de tais sistemas são os telefones celulares com máquina fotográfica e agenda, o sistema de controle dos carros e ônibus, os computadores portáteis *palm-top*, os fornos de micro-ondas com controle de temperatura inteligente, as máquinas de lavar e outros eletrodomésticos.

Os sistemas embarcados também são incorporados na automação industrial com a finalidade de auxiliar nos seus processos. Desta forma, com o objetivo de aumentar a eficiência da indústria de couros, um sistema de visão computacional para classificação automática da qualidade das peças de couro caprino pode ser projetado.

O projeto deste tipo de sistema computacional é extremamente complexo, por envolver conceitos até agora pouco analisados pela computação de propósitos gerais. Por exemplo, as questões da portabilidade e do limite de consumo de potência sem perda de desempenho, a baixa disponibilidade de memória, a necessidade de segurança e confiabilidade, a possibilidade de funcionamento em uma rede maior, e o curto tempo de projeto tornam o desenvolvimento de sistemas computacionais embarcados uma área em si (WOLF, 2016).

Um sistema embarcado pode ser projetado para aplicações que precisam de um sistema de visão computacional, no qual está incluído um processo de classificação de imagens. Segundo Silvia (2007), uma parte essencial do processo de classificação de imagens consiste em convertê-las em vetores, estes de características a serem utilizados no treinamento dos diversos classificadores de padrões, avaliados na tarefa de interesse. Os tipos mais comuns de características ou atributos incluem bordas, contornos, superfícies e outras características salientes tais como esquinas, interseção de linhas e pontos de alta curvatura da imagem.

Os sistemas de visão computacional evoluem gradativamente ao longo dos anos, proporcionando uma redução dos custos devido ao aumento do número de empresas especializadas no desenvolvimento de *software* e hardware específicos para processamento de imagens (SANTOS FILHO, 2013).

De acordo com Amorim (2009), reconhecimento de padrões é basicamente o processo de identificar uma categoria de padrões, baseada em um conjunto de dados. Esta área abrange desde a detecção de padrões até a escolha entre objetos para a classificação, essa baseada em um conjunto de informações. O reconhecimento de padrões em imagens

exige diversos processos computacionais, que ao serem utilizados facilitam seu processo de reconhecimento, como por exemplo, a utilização de pré-processamento, extração e seleção de atributos com a utilização de técnicas de processamento digital de imagens (PDI), entre outras.

As abordagens apresentadas nesta dissertação foram realizadas e testadas nas aplicações dos trabalhos de SANTOS FILHO (2013) e Marinho (2017). A primeira pesquisa descreve o desenvolvimento de um sistema de visão computacional para classificação automática da qualidade das peças de couro de ovinos/caprinos. Utilizou-se diferentes técnicas de extração de características para o reconhecimento dos padrões. A segunda apresenta um estudo comparativo do desempenho de diversos descritores de imagem e classificadores na tarefa de reconhecimento de locais internos para localização e navegação de um robô móvel usando mapa topológico. Vale ressaltar que essas duas aplicações apresentam um problema de classificação multiclases.

Neste trabalho foi utilizado o classificador SVM (support vector machine) em conjunto com dois descritores baseados em texturas. Um que usa a técnica de extração de características de imagens digitais a partir de uma matriz de coocorrência de níveis de cinza (Gray Level Co-Occurrence Matrix, GLCM) e o outro baseado em padrões binários locais (Local Binary Patterns, LBP).

O classificador SVM foi escolhido por ser um dos algoritmos mais eficientes de aprendizado de máquina, e bastante utilizado em reconhecimento de padrões desde a sua introdução na década de 1990 (CORTES; VAPNIK, 1995), por apresentar bons desempenhos em Marinho et al. (2017), por se encontrar poucos trabalhos na literatura (LIN, 2009; SHI; CHAN; LI, 2009; JODAS et al., 2013; OLIVEIRA; WEHRMEISTER, 2016; MEHTA; TOMAR, 2016; SETHUPATHY; Veni S, 2016) que tentaram embarcar esse classificador e por estar disponível na biblioteca OpenCV. Já os métodos de extração de atributos GLCM e LBP foram escolhidos por terem sido largamente utilizados em aplicações na literatura, onde a análise e descrição de texturas das imagens são essenciais para um bom desempenho do processo de classificação.

Sendo assim, este trabalho apresenta estratégias para escolha de atributos que preservam a taxa de acerto (acurácia) do classificador e reduzem o uso de recursos computacionais. Estas estratégias são avaliadas no âmbito de embarcar todo o processo de classificação por imagens de uma aplicação em uma plataforma alvo.

1.1 Motivação

A principal motivação deste trabalho é investigar relações entre as características do processo de classificação de imagens e o consumo de recursos em um computador embarcado. Existem diversas aplicações embarcadas assim como diversos estudos e aplicações de

classificadores para um grande conjunto de problemas em campo (VIEIRA, 2016). Porém, na literatura não foram encontrados trabalhos que apresentassem avaliações sobre como embarcar um sistema de visão computacional, utilizando especificamente o SVM, discutindo suas restrições em uma plataforma embarcada.

Aplicações embarcadas de visão computacional têm sido incorporadas na automação industrial, com o objetivo de melhorar a qualidade e segurança dos processos. Tais sistemas envolvem classificadores de padrões para funções específicas, que muitas vezes apresentam elevado custo para armazenamento e tempo de processamento.

1.2 Hipóteses de Trabalho

As hipóteses básicas estabelecidas para esta dissertação são:

- a resolução da imagem usada pelo classificador influencia a sua capacidade de classificação. Imagens com baixa definição devem aumentar a taxa de erro do classificador;
- a quantidade de atributos gerada pelo extrator influencia a acurácia do classificador. Quanto mais atributos, melhor a taxa de acerto.

1.3 Objetivos Geral e Específicos

O objetivo geral deste trabalho é reduzir custos computacionais para embarcar um sistema de visão computacional. Possibilitando assim um estudo do desempenho em uma plataforma embarcada.

Os objetivos específicos são:

- testar o desempenho dos métodos de extração de características visando sua utilização no sistema de classificação;
- avaliar o tempo de computação do classificador SVM em conjunto com GLCM e LBP para diferentes aplicações;
- Identificar, no processo de classificação por imagens, algumas propriedades que afetam diretamente o consumo de recursos na plataforma embarcada;
- avaliar os resultados obtidos em uma plataforma embarcada.

É importante salientar que este trabalho não se propõe a melhorar a acurácia do classificador. O objetivo não é obter melhores resultados nesse quesito, mas tão somente reproduzir o que outros trabalhos conseguiram, porém com recursos mínimos.

1.4 Relevância

A presente dissertação de mestrado é relevante por tratar de um estudo completo da viabilidade de embarcar os métodos aqui demonstrados, possibilitando ao interessado final tomar decisões no projeto de um classificador embarcado a partir das avaliações das restrições e desempenho.

Quanto às aplicações de visão computacional tratados nesta dissertação, não foram encontrados trabalhos na literatura que avaliassem as restrições e desempenho em alguma plataforma embarcada.

1.5 Problematização

A principal questão subjacente à elaboração desta dissertação é a avaliação do impacto de redução do custo computacional em aplicações embarcadas utilizando o classificador SVM combinado com os extratores de texturas GLCM e LBP, sempre preservando a taxa de acerto do classificador.

1.6 Estrutura da Dissertação

No Capítulo 2, como fundamentação teórica, são apresentados os conceitos básicos utilizados no presente trabalho. São descritos no Capítulo 3 os estado da arte no processo de embarcar o classificador SVM, assim como os extratores de atributos GLCM e LBP. Também são apresentados os estado da arte na classificação de couro e localização e navegação de robôs móveis.

O Capítulo 4 descreve os métodos ou procedimentos abordados nesta dissertação. As duas aplicações adotadas, assim como suas bases de dados são apresentadas. Na sequência o processo de redimensionamento das imagens é detalhado. Logo após, as etapas do processo de segmentação para extrair a região de interesse nas imagens são demonstradas. Em seguida, duas abordagens para reduzir a quantidade de atributos são apresentadas, assim como os conjuntos de atributos avaliados. O restante do capítulo detalha o processo de classificação e avaliação adotado no trabalho. Finalmente, é demonstrado como foi realizado o treinamento do classificador SVM, bem como os testes a partir das métricas também apresentadas.

No Capítulo 5 são apresentados os resultados obtidos nas estações de trabalho e em uma plataforma embarcada para cada aplicação. O Capítulo 6 finaliza o trabalho, onde se apresentam as conclusões da pesquisa, contribuições e trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo são apresentados os principais conceitos utilizados nesse trabalho, com a limitação de abranger o que é mais pertinente ao entendimento dos experimentos realizados. Inicialmente, conceitos sobre processamento digital de imagens são apresentados. No segundo momento, os extratores de características são explicados e serão detalhados os parâmetros utilizados. Em seguida, conceitos sobre reconhecimento de padrões são explorados. Logo após, o classificador adotado nesta dissertação é estudado. Por fim, são apresentados características e conceitos de um projeto de sistemas embarcados.

2.1 Processamento Digital de Imagens - PDI

Processamento Digital de Imagens (PDI) são técnicas voltadas para a análise de dados multidimensionais, adquiridos por diversos tipos de sensores, ou seja, é a manipulação de uma imagem por computador de modo que a entrada e a saída do processo sejam imagens. O objetivo de se usar PDI é melhorar o aspecto visual de certas feições estruturais para o analista humano e fornecer outros subsídios para a sua interpretação, inclusive gerar produtos que possam ser posteriormente submetidos a outros processamentos (BRYN, 2008).

Uma imagem pode ser definida como uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais, e a amplitude de f em qualquer par de coordenadas (x, y) é chamado de intensidade da imagem nesse ponto. Quando x , y e os valores de intensidade de f são quantidades finitas e discretas, chama-se de imagem digital (GONZALEZ; WOODS, 2010). Neste trabalho foi adotado a convenção de par de eixos (x, y) demonstrado na Figura 1.

Várias metodologias podem ser aplicadas a imagens para diferentes propósitos e objetivos (GONZALEZ; WOODS, 2010). A Figura 2 apresenta os passos ou métodos fundamentais utilizados no processamento de imagens digitais. Neste trabalho foram utilizados os métodos de: aquisição de imagens, filtragem e realce de imagens, segmentação, representação e descrição, e reconhecimento de padrões em imagens.

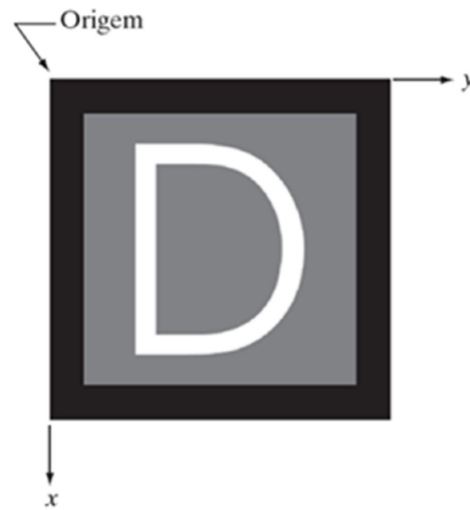


Figura 1 – Imagem representada como uma matriz de intensidade visual (GONZALEZ; WOODS, 2010).

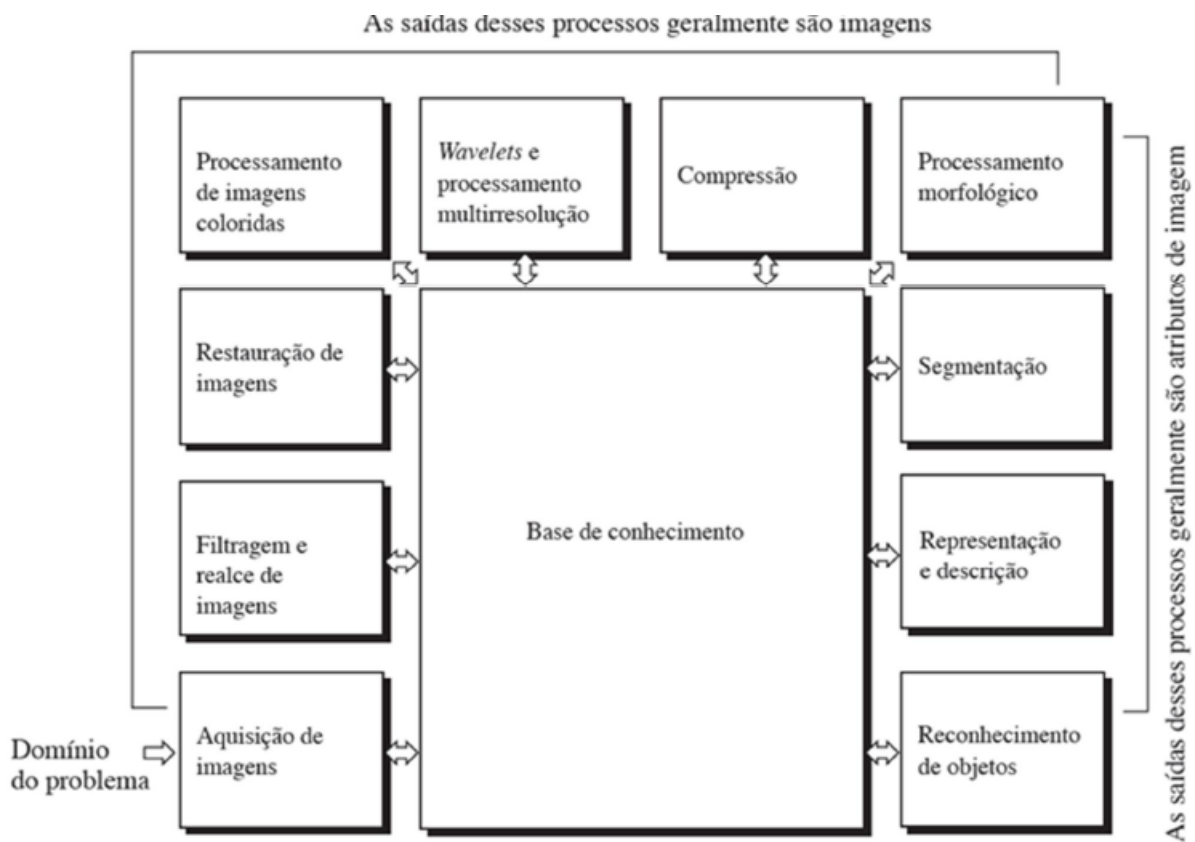


Figura 2 – Passos fundamentais em processamento de imagens digitais (GONZALEZ; WOODS, 2010).

Na área de PDI, visão computacional é o estudo da extração de informação de uma imagem; mais especificamente, é a construção de descrições explícitas e claras dos objetos em uma imagem (BALLARD; BROWN, 1982). Um Sistema de Visão Computacional ou Artificial está caracterizado como um sistema computadorizado capaz de adquirir, processar e interpretar imagens correspondentes a cenas reais (MARQUES FILHO; VIEIRA NETO, 1999). A Figura 3 mostra esquematicamente um diagrama de um sistema de visão computacional com o objetivo de explicar suas principais etapas.

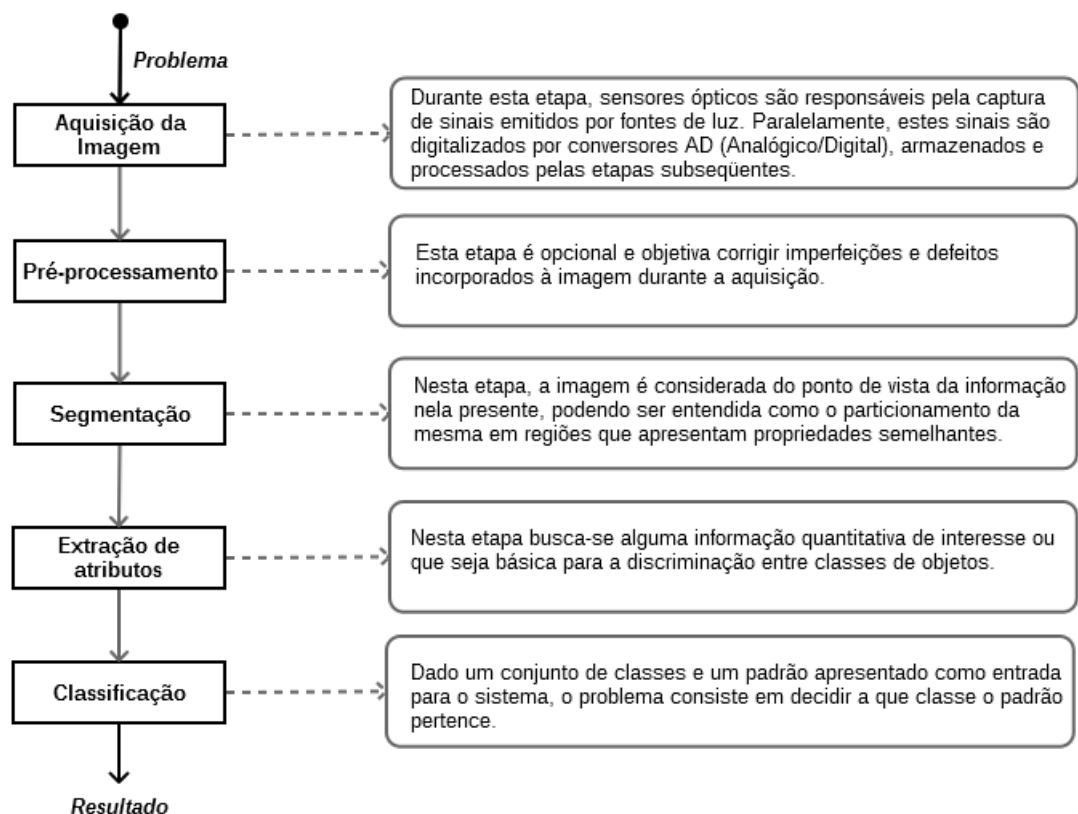


Figura 3 – Etapas de um sistema de visão computacional (MARQUES FILHO; VIEIRA NETO, 1999).

Todas as etapas são atreladas a uma base de conhecimento ou domínio do problema. Nesta dissertação, consiste nos problemas de classificação automática de couros caprinos e localização e navegação de robô móvel.

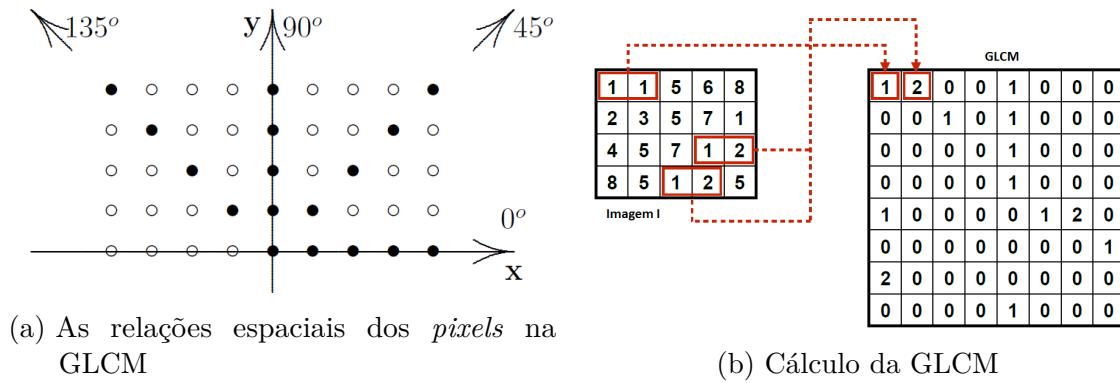
2.2 Extratores de Características

De acordo com Marinho (2017) os descritores de características devem ser robustos e invariantes às mudanças. Além disso, precisam ser extraídos em um tempo razoável e com menor complexidade possível. É importante ressaltar que a escolha do descritor ideal está fortemente ligada à aplicação.

2.2.1 Matriz de Coocorrência de Níveis de Cinza

A Matriz de Coocorrência de Níveis de Cinza (GLCM) é uma técnica que têm sido utilizada em muitos trabalhos para medir a textura de imagens desde que foi proposta em (HARALICK; SHANMUGAM; DINSTEIN, 1973). Na GLCM, cada entrada (i, j) na Matriz de Coocorrência (MC) corresponde ao número de ocorrências do par de níveis de cinza i e j que estão a uma distância D e direção θ na imagem original. A MC é definida da seguinte forma: $p(i, j, D, \theta) = P(i, j, D, \theta)/R$, onde R é uma constante padronizada da soma de cada entrada (i, j) (HARALICK; SHANMUGAM; DINSTEIN, 1973).

A Figura 4a apresenta as características da GLCM. A orientação θ pode ser representada em quatro direções: horizontal, diagonal, vertical e anti-diagonal (0° , 45° , 90° e 135°).



Fonte: (ALBREGTSEN, 2008)

Fonte: (MATHWORKS, 2017)

Figura 4 – Representação da GLCM

Uma MC descreve a quantidade de combinações de níveis de cinza em uma imagem com uma determinada direção e distância entre o *pixels* vizinhos. O número de linhas e colunas dessa matriz é igual ao número de níveis de cinza, N_g , na imagem. Neste trabalho inicialmente foi adotado $N_g = 256$. Na Figura 4b mostra um exemplo para o cálculo dos valores da MC de uma imagem **I** 4×5 *pixels* com $\theta = 0^\circ$. O elemento $(1, 1)$ na MC contém o valor 1 porque só existe uma instância na imagem onde dois *pixels*, horizontalmente adjacentes, têm os valores 1 e 1. O elemento $(1, 2)$ contém o valor 2 porque existem duas instâncias na imagem onde dois pixels, horizontalmente adjacentes, têm os valores 1 e 2.

O trabalhos de Haralick, Shanmugam e Dinstein (1973), Soh e Tsatsoulis (1999) e Wang et al. (2010), juntos disponibilizam um conjunto de 24 medidas estatísticas a partir de uma MC que podem ser usadas para representar características presentes em texturas, extraídas de uma única imagem. Todas essas medidas ou descritores (f_1 a f_{24}), inicialmente, foram adotadas como atributos neste trabalho. Essas medidas são definidas conforme as equações abaixo.

Notação:

- $p(i, j)$: (i, j) -ésima entrada em uma matriz normalizada de dependência espacial de níveis de cinza $p(i, j)/R$;
- $p_x(i)$: i -ésima entrada em uma matriz de probabilidade marginal obtida pela soma das linhas de $p(i, j)$, $\sum_{j=1}^{N_g} p(i, j)$;
- $p_y(j) = \sum_{i=1}^{N_g} p(i, j)$;
- N_g : Número de níveis de cinza na imagem quantizada;
- $\sum_i$ e $\sum_j$: forma simplificada de $\sum_{i=1}^{N_g}$ e $\sum_{j=1}^{N_g}$, respectivamente;
- $p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)$ para $k = 2, 3, \dots, 2N_g$;
- $p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)$ para $k = 2, 3, \dots, 2N_g - 1$;

Atributos:

- Segundo Momento Angular (*Angular Second Moment - ASM*):

$$f_1 = \sum_i \sum_j [p(i, j)]^2. \quad (2.1)$$

- Contraste (*Contrast*):

$$f_2 = \sum_{k=0}^{N_g-1} k^2 [p_{x-y}(k)]. \quad (2.2)$$

- Correlação (*Correlation*):

$$f_3 = \frac{\sum_i \sum_j (i, j) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}, \quad (2.3)$$

onde μ_x , μ_y , σ_x e σ_y são as médias e desvios padrões de p_x e p_y .

- Soma dos quadrados ou Variância (*Sum of Squares or Variance*):

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j). \quad (2.4)$$

- Momento da diferença inversa (*Inverse Difference Moment - IDM*):

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i + j)^2} p(i, j). \quad (2.5)$$

- Média da soma I (*Sum Average*):

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i). \quad (2.6)$$

- Variância da soma (*Sum Variance*):

$$f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i). \quad (2.7)$$

- Entropia da soma (*Sum Entropy*):

$$f_8 = - \sum_{i=2}^{2N_g} \log[p_{x+y}(i)]. \quad (2.8)$$

- Entropia (*Entropy*):

$$f_9 = - \sum_i \sum_j p(i, j) \log [p(i, j)]. \quad (2.9)$$

- Variância da diferença (*Difference Variance*):

$$f_{10} = \text{variância de } p_{x-y}. \quad (2.10)$$

- Entropia da diferença (*Difference Entropy*):

$$f_{11} = - \sum_i^{N_g-1} p_{x-y} \log [p_{x-y}(i)]. \quad (2.11)$$

- Informação das Medidas de correlação (*Information Measures of Correlation - IMC I , IMC II*):

$$f_{12} = \frac{HXY - HXY1}{\max[HX, HY]}, \quad (2.12)$$

$$f_{13} = 1 - \exp[-2.0(HXY2 - HXY)]^{1/2}, \quad (2.13)$$

onde:

- $HXY = - \sum_i \sum_j p(i, j) \log[p(i, j)]$;
- HX e HY são entropias de p_x e p_y ;
- $HXY1 = - \sum_i \sum_j p(i, j) \log[p_x(i)p_y(j)]$;
- $HXY2 = - \sum_i \sum_j p_x(i)p_y(j) \log[p_x(i)p_y(j)]$.

- Coeficiente máximo de correlação (*Maximal Correlation Coefficient - MCC*):

$$f_{14} = (\text{segundo maior autovalor de } \mathbf{Q})^{1/2}, \quad (2.14)$$

onde:

$$\mathbf{Q} = \sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)}. \quad (2.15)$$

- Homogeneidade (*Homogeneity*):

$$f_{15} = \sum_i \sum_j \frac{p(i, j)}{1 + |i - j|}, \quad (2.16)$$

- Média da soma II (*Sum Mean*):

$$f_{16} = \frac{1}{2} \sum_i \sum_j p(i, j)(i + j), \quad (2.17)$$

- Probabilidade máxima (*Maximum Probability*):

$$f_{17} = \max_{i,j} [p(i, j)], \quad (2.18)$$

- Tendência de agrupamento (*Cluster Tendency*):

$$f_{18} = \sum_i \sum_j (i + j - 2\mu)^2 p(i, j), \quad (2.19)$$

- Tons de agrupamento (*Cluster Shade*):

$$f_{19} = \sum_i \sum_j p(i, j)(i + j - \mu_x - \mu_y)^3, \quad (2.20)$$

- Proeminência do agrupamento (*Cluster Prominence*):

$$f_{20} = \sum_i \sum_j p(i, j)(i + j - \mu_x - \mu_y)^4, \quad (2.21)$$

- Dissimilaridade (*Dissimilarity*):

$$f_{21} = \sum_i \sum_j p(i, j) |i - j|, \quad (2.22)$$

- Média da diferença (*Difference Mean*):

$$f_{22} = \frac{1}{2} \sum_i \sum_j p(i, j)(i - j), \quad (2.23)$$

- Autocorrelação (*Autocorrelation*):

$$f_{23} = \sum_i \sum_j (ij)p(i, j), \quad (2.24)$$

- Inércia (*Inertia*):

$$f_{24} = \frac{1}{(N_g^2 - 1)^2} \sum_i \sum_j p(i, j)(i - j)^2, \quad (2.25)$$

2.2.2 Padrões Binários Locais

Os Padrões Binários Locais (Local Binary Patterns, LBP) são operadores de textura simples, porém eficientes, que rotulam os *pixels* de uma imagem ao limitar a vizinhança de cada *pixel* e considera o resultado como um número binário. Devido ao seu poder discriminativo, o operador de textura LBP tornou-se uma abordagem popular em várias aplicações. Pode ser visto como uma abordagem unificadora dos modelos estatísticos e estruturais tradicionalmente divergentes de análise de textura. Outra propriedade importante é a simplicidade computacional, que permite analisar imagens em configurações desafiadoras em tempo real (PIETIKÄINEN, 2010).

O operador LBP original, formulado por Ojala, Pietikainen e Harwood (1994), atribui um rótulo (representado por um número binário) a cada pixel analisado. Para atribuir esse rótulo, é verificado se cada vizinho do *pixel* em análise, em um raio considerado, excede, ou não, um limiar (*threshold*) que é determinado pelo valor do *pixel* central, cujo rótulo está sendo determinado. Cada vizinho que ultrapassar ou se igualar ao limiar, será marcado com o dígito binário 1, caso contrário, receberá 0. A junção desses dígitos binários representará o *pixel* central. Sendo assim, faz-se necessário seguir uma lógica para determinar qual o dígito mais significativo. Em outras palavras, a partir de que vizinho o número começará a ser formado, visto que a vizinhança é circular. Também é preciso observar que quando se determina o dígito mais significativo, automaticamente é distribuído um peso a cada um dos dígitos. A soma da multiplicação de cada dígito pelo seu respectivo peso retornará o que é referenciado como código LBP. Como a vizinhança consiste de 8 *pixels*, um total de $2^8 = 256$ padrões podem ser codificados. Um exemplo da computação do operador LBP original é mostrado na Figura 5.

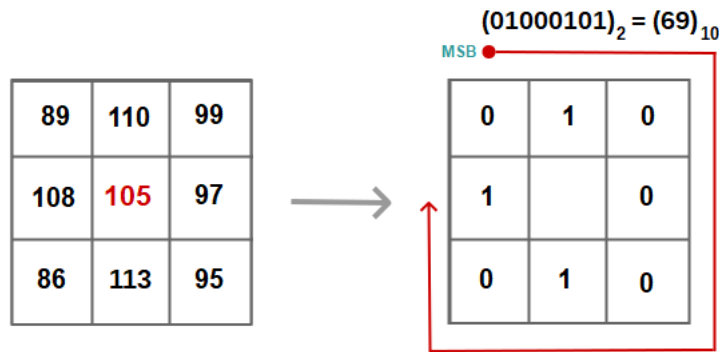


Figura 5 – Exemplo do cálculo do LBP. Para uma vizinhança 3×3 na imagem, o padrão binário local é determinado pela comparação do *pixel* central com seus vizinhos na ordem fixada atribuída pela seta. O padrão binário gera um número que é computado no histograma LBP.

Formalmente, a partir de uma imagem em escala de cinza **I**. Seja g_c o valor da intensidade de um *pixel* qualquer $g_c = \mathbf{I}(x, y)$ e g_p o valor de intensidade na vizinhança espacial 3×3 de (x, y) . O rótulo LBP de padrão de intensidade da vizinhança espacial do pixel (x, y) é

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.26)$$

onde $P = 8$ e $s(\cdot)$ é a função degrau:

$$s(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (2.27)$$

Pietikäinen et al. (2011a) propuseram uma metodologia, denominada Padrões Uniformes (Uniform Patterns), para reduzir, significativamente, a dimensionalidade gerada pelo operador LBP original. Esta extensão foi inspirada pelo fato de que alguns padrões binários ocorrem mais comumente em imagens de textura do que outros. Um padrão binário local é chamado de uniforme se o padrão binário contém no máximo 2 transições de *bit* a *bit* de 0 para 1, ou vice-versa, quando o padrão de bits é percorrido circularmente. Por exemplo, os padrões 00000000 (0 transições), 01110000 (2 transições) e 11001111 (2 transições) são uniformes, enquanto os padrões 11001001 (4 transições) e 01010010 (6 transições) não são. Na computação dos rótulos LBP, padrões uniformes são usados para que haja um rótulo separado para cada padrão uniforme e todos os padrões não uniformes são rotulados com um único rótulo. Portanto, o número de rótulos diferentes para a vizinhança de P *pixels* é $P(P - 1) + 3$. Uma vizinhança 3×3 possui 59 padrões únicos (58 para os padrões uniformes e 1 para os não uniformes). Ojala, Pietikainen e Maenpää (2002) observaram em seus experimentos que uma vizinhança 3×3 de padrões uniformes representa, aproximadamente, 90% de todos os padrões da textura.

2.3 Reconhecimento de Padrões

A área de reconhecimento de padrões é a parte essencial de qualquer sistema de análise de imagens de alto-nível. Para reduzir a dimensão e a redundância dos dados, um conjunto de valores numéricos, conhecidos como atributos/características, é extraído das imagens e utilizado para representá-las. Um sistema de reconhecimento flexível deve ser capaz de reconhecer um objeto independentemente de sua orientação, tamanho e localização no campo de visão (ERPEN, 2004).

Segundo Bishop (2006), o reconhecimento de padrões tem sua origem na engenharia, enquanto o aprendizado de máquina cresceu na Ciência da Computação. No entanto, essas atividades podem ser vistas como duas facetas do mesmo campo, inteligência artificial.

O principal objetivo dessa área de estudo é a classificação de padrões dentro de um número de categorias ou classes. Conforme a aplicação, esses padrões podem ser imagens, formas de onda de sinal ou algum tipo de medição que precisa ser classificada (THEODORIDIS; KOUTROUMBAS, 2008).

O reconhecimento de padrões se aplica a inúmeros problemas de processamento de informações, desde reconhecimento de voz, detecção de erros em equipamentos ou até mesmo, inspeção visual para detecção de imperfeições em texturas (AMORIM, 2009). O projeto de um sistema de reconhecimento de padrões envolve, essencialmente, quatro etapas:

- Aquisição de dados (extração de características);
- Pré-processamento de dados (seleção de características mais discriminativas);

- Representação de dados;
- Tomada de decisão (construção de um classificador).

O tipo de domínio é determinante para a escolha da técnica de pré-processamento, do esquema de representação e do modelo de tomada de decisão. Um problema bem definido e suficientemente detalhado é caracterizado por pequenas variações intra-classes (pequenas variações para padrões da mesma classe) e grandes variações inter-classes (grandes variações entre padrões de classes distintas). Por consequência conduz a uma representação compacta das amostras e simplifica a tomada de decisão (MARINHO, 2017).

As técnicas de aprendizagem de um sistema de reconhecimento de padrões empregam um princípio de inferência denominado indução, no qual obtém-se conclusões genéricas a partir de um conjunto particular de exemplos (LORENA; CARVALHO, 2007). O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não-supervisionado.

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada (HAYKIN, 1999). O algoritmo de aprendizagem extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

No aprendizado não-supervisionado não há a presença de um professor, ou seja, não existem exemplos rotulados. O algoritmo de aprendizagem tenta representar (ou agrupar) as entradas submetidas segundo uma medida de qualidade.

Neste trabalho será utilizado o tipo de aprendizado supervisionado. Neste caso, dado um conjunto de exemplos rotulados na forma $(\mathbf{x}_i, y_i)$, em que $\mathbf{x}_i$ representa um exemplo e y_i denota o seu rótulo, deve-se produzir um classificador, também denominado modelo, preditor ou hipótese, capaz de predizer o rótulo de novos dados. Esse processo de indução de um classificador a partir de um conjunto de dados é denominado treinamento. O classificador obtido também pode ser visto como uma função f , a qual recebe um dado $\mathbf{x}$ e fornece uma predição y .

Os rótulos ou classes assumem valores discretos $1, \dots, k$, para $k > 1$. Um problema de classificação no qual $k = 2$ é denominado binário. Para $k > 2$, configura-se um problema multiclasse.

A Figura 6 apresenta de forma simplificada, os conceitos referentes à geração de um classificador a partir do aprendizado supervisionado. Tem-se nessa figura um conjunto com n padrões. Cada padrão $\mathbf{x}_i$ possui m atributos, ou seja, $\tilde{\mathbf{x}}_i = [x_{i1}, \dots, x_{im}]^T$. As variáveis y_i representam as classes. A partir dos exemplos e as suas respectivas classes, o algoritmo de aprendizagem extrai um classificador.

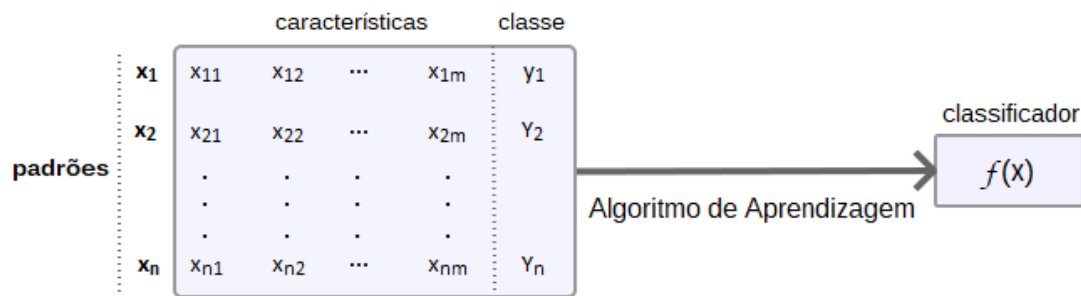


Figura 6 – Indução de classificador em aprendizado supervisionado (LORENA; CARVALHO, 2007).

2.4 Máquina de Vetores Suporte

A Máquina de Vetores Suporte (Support Vector Machine - SVM) é cada vez mais utilizada na solução de diversos tipos de problemas, pois têm apresentado resultados equivalentes a outros algoritmos de Aprendizagem de Máquinas, como as Redes Neurais Artificiais (RNAs) (LORENA; CARVALHO, 2007). A SVM é uma técnica embasada na Teoria de Aprendizado Estatístico (TAE), que consiste em estabelecer condições matemáticas que permitem escolher um classificador, com bom desempenho, para o conjunto de dados disponíveis para treinamento e teste (VAPNIK, 1995).

O processo de indução de classificadores usado em SVM é supervisionado. Desta forma, o processo de aprendizagem é realizado com base nos diversos pares de entrada e saída pertencentes à base de dados de exemplos (ROCHA NETO, 2011).

A SVM, originalmente desenvolvida para classificação binária, busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre os exemplos seja máxima. Isso com padrões linearmente separáveis ¹.

Um conjunto de padrões é dito não-linearmente separável, caso não seja possível separar os dados com um hiperplano. A Figura 7 mostra um conjunto linearmente e outro não-linearmente separável.

¹ Os padrões devem estar suficientemente separados entre si para assegurar que a superfície de decisão consista de um hiperplano (HAYKIN, 1999).

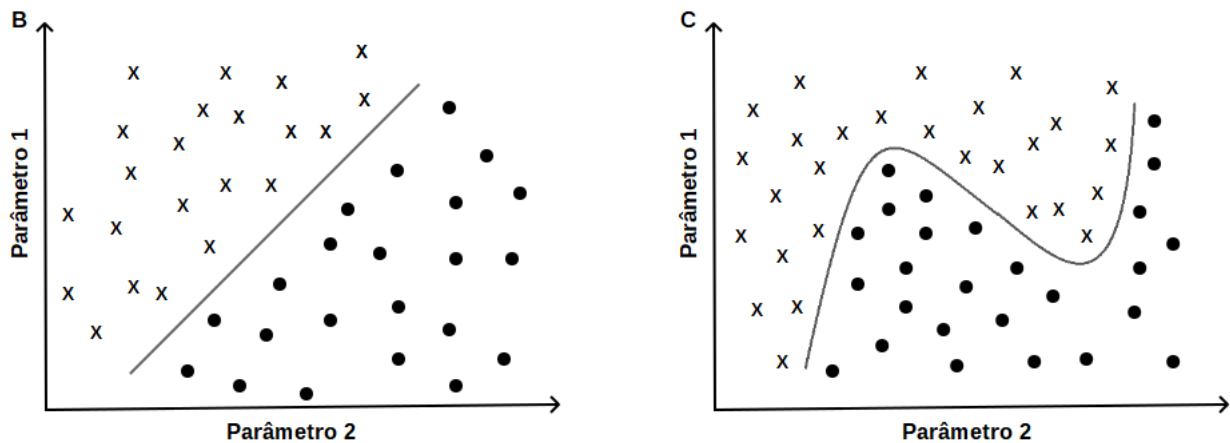


Figura 7 – Exemplos de padrões linearmente e não-linearmente separável respectivamente (GONCALVES, 2015).

Para os padrões não-linearmente separáveis, busca-se uma função de mapeamento (*Kernel*) apropriada para tornar o conjunto mapeado em linearmente separável. Desta forma, pode-se trabalhar indiretamente em espaço de elevada dimensão, desde que as computações sejam realizadas em um outro espaço, denominado Espaço *Kernel* (*Kernel Space*) (ROCHA NETO, 2011). Com isso, a SVM não-linear realiza uma mudança de dimensionalidade, caindo então em um problema de classificação linear, com a opção de fazer uso do hiperplano ótimo ² (GONCALVES, 2015).

Os classificadores SVM podem usar diferentes *kernels* em seu processo de tomada de decisão. De acordo com (ROCHA NETO, 2011) os mais comumente utilizados são o linear, polinomial e gaussiano (RBF) apresentados na Tabela 1. Assim, seus respectivos parâmetros provocam alterações nos resultados fornecidos por um classificador SVM.

Tabela 1 – Resumo dos *kernels* mais populares (ROCHA NETO, 2011).

<i>Kernel</i>	Descrição
Linear	$K(x, x_i) = x_i^T \cdot x$
Polinomial	$K(x, x_i) = (x_i^T \cdot x + 1)^d$, em que d é o grau do polinômio.
Gaussiano (ou RBF)	$K(x, x_i) = \exp \left\{ -\ x - x_i\ ^2 / \sigma^2 \right\}$, em que σ é uma constante.

Sendo assim, algumas escolhas devem ser feitas para obtenção de um classificador SVM, como a função *kernel*, os parâmetros da função, além da definição do algoritmo para determinação do hiperplano ótimo.

² Hiperplano de separação com maior margem. Entende-se como margem, a menor distância entre os exemplos do conjunto de treinamento e o hiperplano utilizado para separação destas classes (LORENA; CARVALHO, 2007)

A base teórica das SVM é bem estabelecida dentro da Matemática e Estatística. Um estudo mais aprofundado pode ser realizado em Smola et al. (2000), Hastie, Tibshirani e Friedman (2008) e em ROCHA NETO (2011).

2.4.1 Classificação Multiclasse

As SVM foram propostas inicialmente para classificação binária. Porém, muitos dos problemas reais são multiclasse, o que torna o problema de aprendizagem consideravelmente mais complexo (ROCHA NETO, 2011). Para que fosse possível a sua utilização neste tipo de aplicação, foram propostos alguns procedimentos para estender um classificador SVM para tratar problemas multiclasse.

2.4.1.1 Abordagem Um-Contra-Todos

O método Um-Contra-Todos (UCT) baseia-se na construção de k classificadores binários, sendo k o número de classes. Cada classificador f_i é responsável por classificar uma classe i das demais (DUAN; KEERTHI, 2005).

Dado um novo padrão x , a classe a qual este novo padrão pertence é a classe representada pelo classificador que obteve o valor máximo entre os k classificadores. Formalmente foi definido em Goncalves (2015) por

$$f(x) = \arg \max_{1 \leq i \leq k} (f_i(x)). \quad (2.28)$$

2.4.1.2 Abordagem Todos-Contra-Todos

Proposto por Friedman (1996) a abordagem Todos-Contra-Todos (TCT) consiste em comparar as classes duas a duas, sendo necessárias $k(k-1)/2$ classificadores, onde k é o número de classes. Para decidir a qual classe pertence um novo padrão x , utiliza-se um esquema de votação por maioria, onde cada uma das SVM fornecem um resultado. A solução final é dada pela classe com maior quantidade de votos (DUAN; KEERTHI, 2005).

Com um grande número de SVM executadas, um alto tempo de processamento é necessário, comparado ao método de decomposição UCT (GONCALVES, 2015). Por exemplo, se $k = 7$, no método UCT são necessárias 7 SVM enquanto no TCT seriam necessárias 21 SVM.

Em relação a taxa de acerto, não foram encontradas na literatura pesquisas que elegeisse uma melhor abordagem, UCT ou TCT. Entretanto, no estudo de Duan e Keerthi (2005) o desempenho do classificador SVM para as abordagens UCT e TCT, para cinco bases de dados diferentes, foram similares.

2.5 Projeto de Sistemas Embarcados

Um Sistema Embarcado é qualquer dispositivo que inclui um computador programável, mas não se destina a ser um computador de propósito geral. Os sistemas computacionais embarcados estão presentes em praticamente todas as atividades humanas (WOLF, 2016). Para Marwedel (2010) Sistemas Embarcados são sistemas de processamento de informações que são incorporados em um produto maior e que normalmente são transparentes para o usuário. Exemplos de sistemas embarcados incluem sistemas de processamento de informação em equipamentos de telecomunicações, sistemas de transporte, equipamentos de fabricação e eletrônicos de consumo.

Para Pimentel (2017), à medida que os sistemas embarcados se tornam mais complexos e, com novas aplicações, exigem muitas restrições de projeto. Sendo assim, o uso de técnicas sofisticadas de exploração do espaço de projeto (design space exploration, DSE) são essenciais para encontrar a melhor harmonia entre os diferentes objetivos de projeto e seus conflitos de decisão.

Durante a DSE, uma grande variedade de projetos diferentes pode ser explorado, como a quantidade e o tipo de processadores implantado na plataforma, e também o tipo da rede de interconexão usada para conectar o sistema de componentes. Além disso, múltiplos objetivos de otimização, como desempenho, consumo de energia e custos, devem ser considerados simultaneamente. Como os objetivos geralmente estão em conflito, pode não haver uma solução ideal que simultaneamente otimize todos os objetivos. Assim sendo, decisões ótimas devem ser tomadas na presença de compensações entre os critérios de projeto. Logo, este processo de DSE pode influenciar fortemente o sucesso ou o fracasso do produto final (PIMENTEL, 2017).

Segundo Wolf (2016), os projetistas devem ser capazes de identificar onde os microprocessadores podem ser usados, projetar uma plataforma de *hardware* com dispositivos de entrada e saída que possa suportar as tarefas requeridas, e implementar um *software* que execute os processos requeridos. Dois dos principais desafios para um projeto de sistemas embarcados são saber quanto de *hardware* será necessário e como minimizar o consumo de energia. No *hardware* pode-se selecionar o tipo de microprocessador usado, a quantidade de memória, os dispositivos periféricos, entre outros.

Para a aplicação deste trabalho, busca-se o menor *hardware* combinado com um *software* que resolva todo o processo de classificação. Neste caso, será observado no fluxo dos dados (extração de atributos e classificação) onde está o maior consumo de recursos (tempo de computação e memória) e como otimizá-los. Sendo assim, com esta pesquisa é possível tornar o processo de projetar um sistema embarcado mais rápido, com a definição de algumas diretrizes para facilitar a tomada de decisão por parte dos projetistas.

3 Trabalhos Relacionados

Neste Capítulo são descritos os trabalhos relacionados que embarcaram o classificador SVM em alguma plataforma. Pesquisas que embarcaram os extratores de atributos GLCM e LBP e os que desenvolveram algum processo aplicado, tanto na classificação de couro, quanto na localização e navegação de robôs móveis.

3.1 Estado da Arte no processo de embarcar o classificador SVM

Diversos trabalhos na literatura utilizaram o SVM em várias aplicações, mas poucos tentaram embarcar esse classificador. No trabalho de Oliveira e Wehrmeister (2016) foi proposto um método para reconhecimento de pessoas em imagens aéreas obtidas a partir de Veículos Aéreos Não Tripulados de pequeno porte com a utilização de Sistemas de Reconhecimento de Padrões (SRP) aplicados ao reconhecimento de imagens. Foram testadas quatro técnicas de aprendizado de máquina, dentre eles o SVM em conjunto com o extrator de características Histogram of Oriented Gradients (HOG). No trabalho foram avaliadas as taxas de acerto dos SRPs, além do seu tempo de processamento em um sistema embarcado de baixo custo e em uma Base de Controle Móvel (BCM). O sistema embarcado adotado nos testes foi a Raspberry Pi 2 *Model B* v1.1. Apesar dos resultados, o autor salientou que o código utilizado não foi otimizado para os recursos de *hardware* da plataforma embarcada já que a utilização de funcionalidades específicas pode reduzir o tempo de processamento.

Um sistema de navegação baseado em algoritmos de processamento de imagens e máquinas de vetores de suporte foi apresentado em Jodas et al. (2013), o qual seria implementado em *hardware* (FPGA), para garantir a dirigibilidade de um robô móvel por caminhos de plantações. O sistema de navegação foi utilizado para manter o robô na trilha da plantação e controlar sua direção baseando-se em imagens do terreno, as quais são utilizadas por algoritmos de processamento de imagens que são aplicados para a melhoria da qualidade da imagem e na extração das características do caminho. O caminho identificado é utilizado por uma máquina de vetores de suporte para a determinação do ângulo de direção do robô móvel. Em um dos testes, baseado apenas nas possíveis limitações de um FPGA, foi simulado em um computador com 4 *Gibabytes* de memória RAM e um processador *Intel Dual Core* de 2200 MHz, o qual apresentou uma precisão em torno de 93% da rota adequada. Entretanto, o trabalho deixou como desafio e possíveis trabalhos futuros uma implementação em *hardware* de uma máquina de vetores de suporte embarcada, assim como uma avaliação das restrições de desempenho.

Em Shi, Chan e Li (2009), foi desenvolvido um sistema para humanos na proteção

contra quedas, com a finalidade de reduzir sua força no impacto. Uma versão do SVM foi implementada e embarcada em uma plataforma com a utilização de um microcontrolador DSP. O SVM foi utilizado para discriminar os movimentos normais dos que caracterizam quedas.

No trabalho de Mehta e Tomar (2016), o SVM foi utilizado em um sistema que realiza as presenças automáticas de alunos em sala de aula, baseado no reconhecimento de faces. Para a captura das imagens foi utilizada uma câmera integrada à placa Raspberry PI 2. O processo de reconhecimento foi realizado na ferramenta Matlab.

Lin (2009) usou o SVM para desenvolver um sistema de visão de máquina para identificação de ervas daninhas em plantações. O processo de classificação foi embarcado em um *laptop* utilizando o *software* MatLab. O sistema apresentou uma acurácia de 95%. Também na agricultura, Sethupathy e Veni S (2016) foi apresentado um estudo para classificar e identificar as doenças das folhas de manga para a agricultura indiana. A classificação e identificação da doença foi realizada utilizando o classificador SVM implementado com a biblioteca OpenCV. O autor deixou como trabalho futuro, a utilização de um *hardware* para embarcar todo o sistema.

3.2 Extratores de atributos GLCM e LBP Embarcados

Uma matriz de coocorrência é usada para a extração de atributos ou características na classificação de imagens em várias pesquisas publicadas. Entretanto, falta um estudo de viabilidade para embarcar essa técnica junto com um classificador.

No trabalho de Sieler, Tanougast e Bouridane (2010), o autor afirma que a principal desvantagem do uso da GLCM é que ele exige um longo tempo de computação, especialmente para imagens muito grandes. Sendo assim, o autor propôs uma arquitetura baseada em FPGA para calcular eficientemente a matriz de coocorrência e extrair 6 atributos (ASM, Média, Variância, Correlação, Contraste e Dissimilaridade). Ainda com o mesmo objetivo, outros trabalhos Tahir et al. (2004), Bariamis et al. (2004), Lopez-Estrada e Cumplido (2006), Iakovidis, Maroulis e Bariamis (2007) propuseram outras arquiteturas, sempre baseadas em FPGA. Nota-se que estas soluções convergem para o uso de paralelismo para diminuir o tempo de computação, a maioria delas suprimindo alguns dos atributos do GLCM.

O LBP, devido à sua simplicidade computacional e poder discriminativo, se tornou popular em várias aplicações com visão computacional (PIETIKÄINEN et al., 2011a). Diversos trabalhos propuseram variações na sua implementação visando diminuir custos computacionais na extração dos atributos. Porém, também não foi encontrado um estudo de viabilidade e desempenho dessa técnica junto com um classificador em uma plataforma embarcada.

Em López et al. (2017), para identificar as soluções que aumentam o desempenho da extração de atributos LBP em tempo real, foram comparadas diferentes implementações em termos de desempenho computacional e consumo de energia. O autor testou o LBP em várias plataformas como por exemplo em unidades de processamentos gráficos (Graphics Processing Unit, GPU) e processadores móveis. Contudo, o trabalho não utilizou um classificador para verificar o desempenho da taxa de acerto utilizando os atributos extraídos.

No trabalho de Kryjak, Komorkiewicz e Gorgon (2012) um sistema implementado em FPGA para detecção de ombros-cabeça de pedestres é apresentado. Foram utilizados o extrator de atributos LBP e SVM para o processo de classificação. O sistema final foi implementado em um Xilinx Virtex 6 FPGA, capaz de processar um fluxo de vídeo de resolução 640x480@60 *fps* em tempo real.

Na busca por outra abordagem, este trabalho propõe investigar soluções para escolha de atributos que preservem a taxa de acerto (acurácia) do classificador a fim de utilizar plataformas baseadas em microcontroladores para aplicações de visão computacional, reduzindo custo de desenvolvimento e de implantação.

3.3 Estado da Arte na classificação de peças de couro

Na pesquisa de Amorim (2009), o autor apresentou uma análise do desempenho em diferentes situações práticas para um problema de classificação binária, na detecção de defeitos em couro bovino. No estudo, a quantidade de atributos foi reduzida utilizando técnicas a partir da análise de discriminante de Fisher. Através das técnicas implementadas foram realizadas diversos experimentos e análises no desempenho de classificação. Em um dos experimentos foi realizado a classificação de defeitos do couro bovino no estágio Wet Blue ¹. Nos desempenhos dos classificadores, apresentados na Tabela 2, a Sequential Minimal Optimization (SMO), que é um algoritmo de treinamento para SVM proposto por (PLATT, 1998), obteve uma taxa de acerto de 84.63% a qual foi superada em desempenho pelos classificadores C4.5 e IBk respectivamente.

Tabela 2 – Resultado classificação imagens Wet Blue (AMORIM, 2009).

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

¹ Consiste no processo de transformação de peles de animais em um material estável, durável e resistente ao ataque de microrganismos. Somente após esse processo, o material leva o nome de couro.(JUNIOR, 2015)

Já no trabalho de SANTOS FILHO (2013), foram avaliados os resultados de um sistema baseado em visão computacional cujo objetivo foi classificar automaticamente a qualidade da pele de caprinos/ovinos. Para isso, comparou-se os desempenhos dos classificadores linear dos mínimos quadrados (least squares classifier, LS), máquina de aprendizado extremo (extreme learning machine, ELM), classificador gaussiano linear regularizado (regularized linear Gaussian classifier, RLGC) e máquinas de vetores suporte (support vector machine, SVM). Os classificadores foram combinados com diversas técnicas de extração de características, tais como a Variância das Colunas (VAR), Transformada Wavelet de Haar (HAAR), Fatoração em Matrizes Não-Negativas (NMF), Análise de Componentes Principais (PCA) e Matrizes de Coocorrência de Níveis de Cinza (GLCM). Para cada método de extração, as imagens originais (3264x2448)*pixels* foram convertidas para níveis de cinza com resolução de 8 bits e redimensionadas em quatro diferentes tamanhos (40x40, 60x60, 80x80 e 100x100)*pixels*. Em seguida, após determinados os vetores de características foi feita a classificação. No trabalho foi construído um banco de 154 imagens de peles de cabras no estágio Wet Blue, com 22 imagens por classe. No ambiente real existiam 7 classes, mas, na pesquisa foram agrupadas em Classe Superior (Classes: 1, 2, 3, 4 e 5) e Classe Inferior (Classes: 6 e 7), ou seja, o problema foi binarizado para melhorar o desempenho do classificador. Contudo, a pesquisa não apresentou um estudo de embarcar todo o sistema em uma plataforma alvo.

A Tabela 3 apresenta os resultados de desempenhos dos classificadores combinados com os extratores, com a utilização do software Matlab R2013a como ambiente computacional no estudo. Para uma possível comparação com esta pesquisa, esses resultados foram atualizados com a utilização da mesma base de imagens de couro caprino desta dissertação. Pois a mesma apresenta uma quantidade maior de amostras por classe.

Tabela 3 – Resultados de SANTOS FILHO (2013) para desempenho dos classificadores com diversos métodos de extração de características (Imagens tamanho 100x100).

Classificador	Extratores de Características				
	VAR	NMF	HAAR	PCA	GLCM
LS	72,00 ± 10,55	72,44 ± 11,97	74,11 ± 9,63	50,00 ± 11,41	70,11 ± 10,79
ELM	70,11 ± 11,84	71,35 ± 10,52	86,73 ± 8,04	55,67 ± 12,06	73,97 ± 10,44
RLGC	73,00 ± 0,70	76,00 ± 0,60	73,70 ± 0,78	72,76 ± 0,55	80,50 ± 0,55
SVM	74,00 ± 7,00	75,70 ± 7,80	81,59 ± 4,32	72,47 ± 6,29	69,10 ± 7,17

3.4 Estado da Arte na localização e navegação de robôs móveis

Estudos atuais aplicados na navegação de robôs móveis baseados em visão computacional estão cada vez mais crescentes.

O trabalho de Bessa et al. (2015) apresenta uma comparação de métodos de localização para robôs móveis através de Redes Neurais Artificiais em imagens omnidirecionais.

O algoritmo para classificação dos padrões foi baseado na rede neural artificial (RNA) MLP. O desempenho do classificador no ambiente real teve taxas de acerto médio próximas a 100% com a utilização dos extratores GIST e PCA. O robô utilizado foi equipado com uma câmera omnidirecional e um *laptop* acoplado para executar os algoritmos utilizados.

Uma nova abordagem de classificação, com opção de rejeição para localização e navegação de robôs móveis em mapas topológicos, com o uso de imagens convencionais e omnidirecionais foi proposta em Marinho (2017). A pesquisa apresentou um estudo comparativo do desempenho de diversos descritores de imagem e classificadores na tarefa de reconhecimento de locais internos, para localização e navegação de um robô móvel usando mapa topológico, com a avaliação de dois tipos de câmera, convencional e omnidirecional. Na metodologia adotada foram avaliadas combinações de extratores de atributos e classificadores promissores existentes na literatura, com a finalidade de reconhecimento de cenas para definir a melhor configuração para cada tipo de câmera. A melhor acurácia obtida com a câmera convencional foi 99,96% através da combinação da Máquina de Aprendizagem Mínima (Minimal Learning Machine with Nearest Neighbors, MLM-NN) com os Momentos Espaciais (ME). Em relação ao desempenho da combinação do SVM com GLCM e LBP, obtiveram acurácias em torno de 99%. Na aplicação foi utilizado um robô equipado com uma das câmeras que se comunicava com um microcomputador por telemetria.

As pesquisas aplicadas na localização e navegação de robôs móveis, deixaram como trabalhos futuros: uma avaliação da complexidade dos algoritmos utilizados, aplicar técnicas de seleção de atributos dos extratores, assim como investigar a influência da resolução das imagens. Todas essas questões buscam diminuir recursos computacionais para viabilizar uma aplicação concreta de visão computacional em um robô móvel e autônomo. Portanto, essa dissertação visa a preencher essas lacunas.

3.5 Considerações

Esta dissertação focou em abordagens para reduzir o custo computacional em aplicações embarcadas. Foram utilizados o classificador SVM e extratores de atributos de imagens por textura, GLCM e LBP. Sendo assim, foram avaliados fatores como tamanho da imagem e quantidade de atributos, além de outros fatores específicos de cada método de extração de atributos.

Diferentemente dos trabalhos relacionados, este trabalho apresenta possíveis soluções para embarcar todo um processo de classificação por imagens em uma plataforma alvo com *hardware* reduzido.

4 Metodologia

Neste capítulo estão descritos os procedimentos e técnicas utilizadas no desenvolvimento da pesquisa nas estações de trabalho utilizando Matlab e Opencv. Inicialmente, as duas aplicações adotadas na pesquisa são apresentadas, assim como suas bases de dados. Em seguida, o processo de redimensionamento das imagens e o processo de segmentação para extrair a região de interesse nas imagens são descritos. Por último, as técnicas de redução de atributos e o processo de classificação (treino e teste), assim como as métricas de avaliação são explanadas.

A Figura 8 apresenta a sequência para a condução dos procedimentos. As etapas são: as definições das bases de dados, processo de redimensionamento das imagens, os processos de segmentação das imagens, extração de atributos, seleção e redução de atributos e o processo de classificação e avaliação. Uma relação contendo todos os conjuntos de atributos utilizados neste trabalho é apresentada.

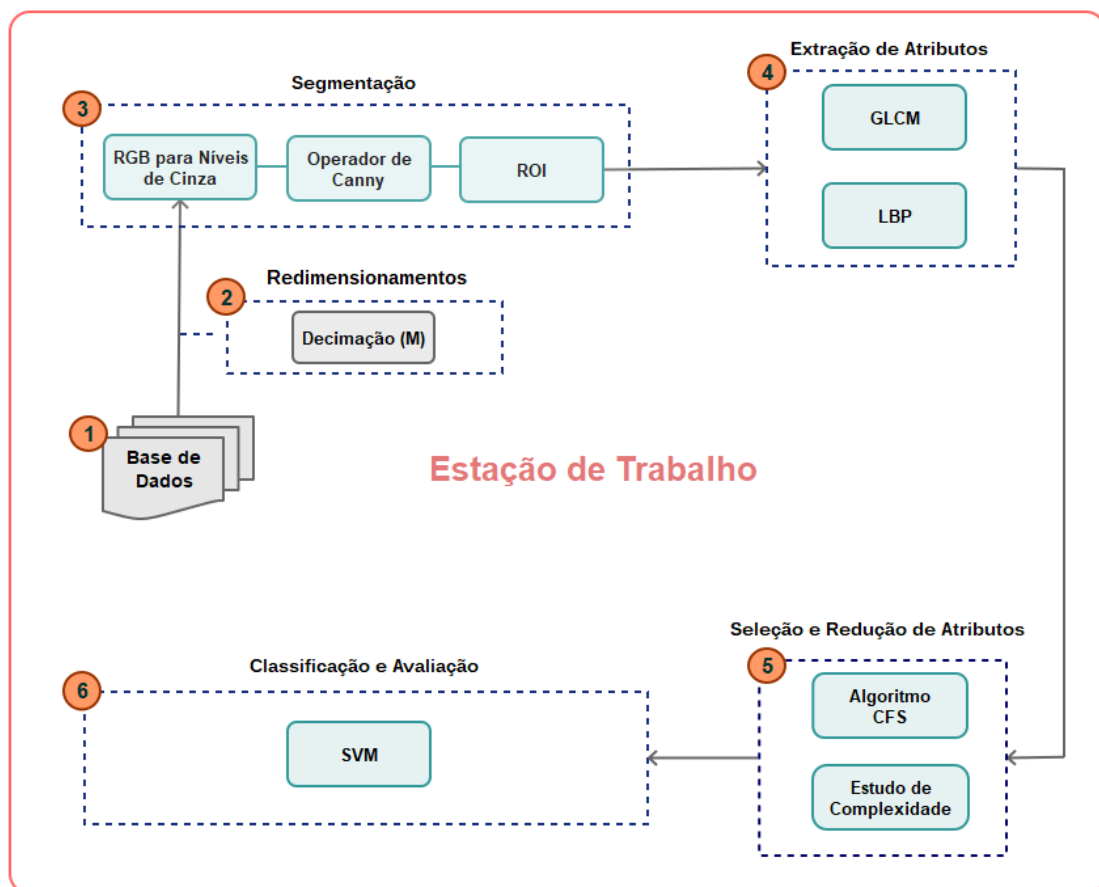


Figura 8 – Fluxo dos procedimentos adotados neste trabalho.

Para implementar os procedimentos e validar os testes realizados neste trabalho, duas estações de trabalho foram utilizadas:

Estação de Trabalho 1 - ET1: Utilizando o software Matlab[®] R2015a, com o intuito de validar os resultados obtidos em SANTOS FILHO (2013) e Marinho (2017). Sua configuração é constituída por um computador Intel[®] Core[™] com 2.20 GHz e 8GB de RAM, rodando o sistema operacional Windows 10 Home Single Language.

Estação de Trabalho 2 - ET2: Utilizando a biblioteca Opencv 2.4.13 em um computador MacBook Air Intel[®] macOS Sierra v10.12.6, Core[™] i5 1.6 GHz, 8GB RAM. A plataforma embarcada alvo utiliza OpenCV e a estação ET2 serve como validação intermediária.

4.1 Aplicações

Neste trabalho duas aplicações distintas de visão computacional foram utilizadas para realizar e testar os procedimentos que buscam reduzir o consumo de recursos computacionais. A primeira aplicação destina-se à classificação automática da qualidade das peças de couro de ovinos/caprino e a outra para localização e navegação de um robô móvel usando mapa topológico. É importante destacar que essas aplicações apresentam um problema de classificação multiclass.

4.1.1 Classificação automática de couro caprino – AP1

Atualmente, o Brasil é um dos maiores produtores mundiais de couros, com capacidade para processar cerca de 45 milhões de unidades por ano. No que diz respeito à criação de ovinos e caprinos, a atividade vem se desenvolvendo em larga escala nos últimos anos, comprovando as previsões de especialistas sobre o potencial dessa atividade. Pesquisa do Instituto Brasileiro de Geografia e Estatística (IBGE) demonstra que o rebanho nacional de ovinos e caprinos já supera 30 milhões de cabeças. No Brasil, a maior concentração dos rebanhos está nas regiões Nordeste, com cerca de 50% dos ovinos e 90% dos caprinos (FURLANETTO, 2007). Contudo, devido ao modo extensivo de criação bem como à informalidade dos abates, as peles chegam aos ambientes industriais com diferentes tipos de defeitos, o que produz peças com diferentes níveis de qualidade. Sendo assim, SANTOS FILHO (2013) propôs um sistema de visão computacional para classificação automática da qualidade das peças de couro de ovinos/caprinos.

4.1.1.1 Base de Dados - BD1

A base de dados BD1 da aplicação AP1 consiste em um banco de 350 imagens de peças de couro caprino no estágio Wet Blue com dimensões de 3456×4608 *pixels*. As

imagens foram obtidas em parceria com a indústria processadora de couros caprinos e ovinos CV Couros e Peles, instalada no estado do Ceará. O processo de aquisição das imagens está descrito em SANTOS FILHO (2013).

Para cada uma das 7 classes foram obtidas 50 imagens. A classe de número 1 apresenta melhor qualidade e a de número 7 apresenta uma qualidade inferior (MEDEIROS, 2007). A Figura 9 exemplifica a classe do tipo 1 e 7 respectivamente.

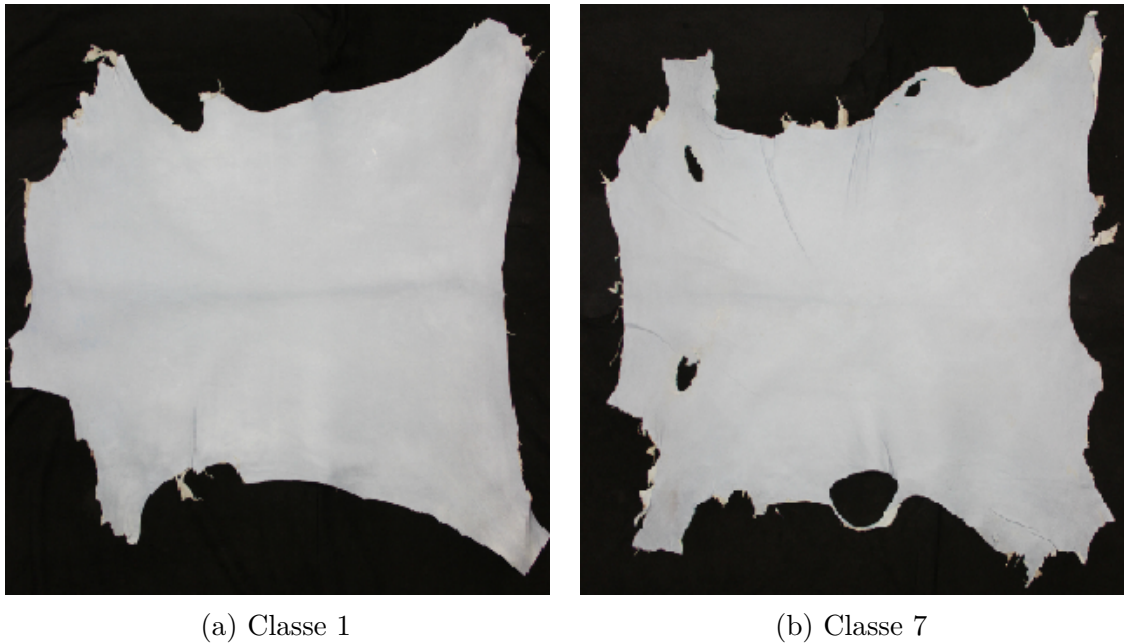


Figura 9 – Exemplos de amostras de imagens das classes do tipo 1 e 7 usadas na aplicação AP1.

As peças de couros classificadas como pertencentes à Classe 1, praticamente não apresentam defeitos visíveis ao olho humano, sendo assim a classe que possui maior valor comercial.

4.1.2 Localização e navegação de robôs móveis em mapas topológicos utilizando imagens convencionais – AP2

A localização do robô móvel, que permite que um robô identifique sua posição, é um dos principais desafios no campo da robótica. Os robôs móveis têm sido amplamente aplicados a tarefas militares e civis. O interesse em robôs móveis autônomos baseados em visão computacional está aumentando, juntamente com os avanços na tecnologia (MARINHO et al., 2017). Posto isto, Marinho (2017) apresentou uma abordagem de navegação e localização de robôs móveis em locais internos.

4.1.2.1 Base de Dados - BD2

A base de dados BD2 da aplicação AP2 é a mesma utilizada e obtida em Marinho (2017). A base consiste em um banco de 600 imagens obtidas no ambiente real com uma câmera convencional. As imagens representam um ambiente interno, no caso um apartamento. Nesse ambiente um mapa topológico foi elaborado para auxiliar no reconhecimento e na navegação do robô. No mapa topológico, ilustrado na Figura 10, os nós, os quais representam os locais no apartamento, são numerados de 1 a 6. A partir desses nós, 15 classes que representam pontos estratégicos e indicam a orientação dentro de um determinado local foram denominadas. Para cada classe foram obtidas 40 imagens com dimensões de 4000×3000 pixels. A Figura 11 exemplifica as classes do tipo 1 e 15 respectivamente.

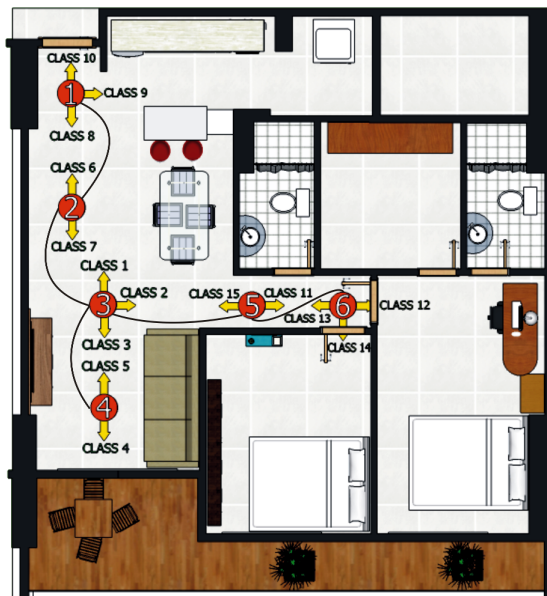
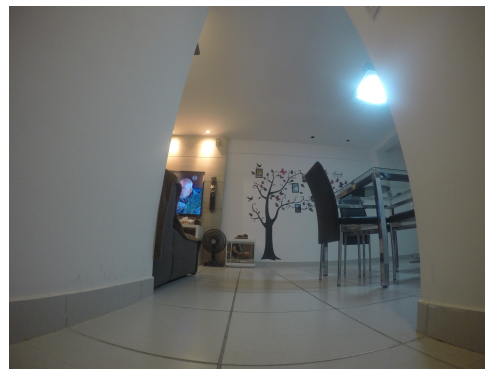


Figura 10 – Mapa topológico do ambiente de navegação (MARINHO, 2017).



(a) Classe 1



(b) Classe 15

Figura 11 – Exemplos de amostras de imagens das classe do tipo 1 e 15 usadas na aplicação AP2.

4.2 Redimensionamento das Imagens

Para verificar o impacto que o tamanho das imagens possa causar no sistema, foram realizados testes no classificador com diferentes redimensionamentos das imagens de entrada. Esse estudo também vai permitir a escolha do tamanho de imagem ideal para ser usado na plataforma embarcada.

O redimensionamento das imagens foi feito usando a técnica de redução da taxa de amostragem de sinais conhecida como decimação. Essa técnica diminui a taxa de amostragem de um sinal mantendo a primeira amostra em cada sequência de conjuntos com M amostras (LYONS, 2010). No caso deste trabalho, sinal equivale a uma imagem e as amostras correspondem aos *pixels* da imagem.

A título de ilustração, a Figura 12 apresenta um exemplo do processo de decimação de uma imagem I (6×8) com $M = 3$, resultando em uma outra imagem J (2×3). Quanto maior o valor de M , menor será a nova imagem. Neste exemplo, somente as linhas 1 e 4 da imagem aparecem na imagem resultante. No primeiro conjunto de amostras da linha 1, foi mantida a amostra 4, no segundo a amostra 2 e por último a amostra 1. Saltando 3 linhas, foram mantidas as amostras 3, 8 e 7 da linha 4.

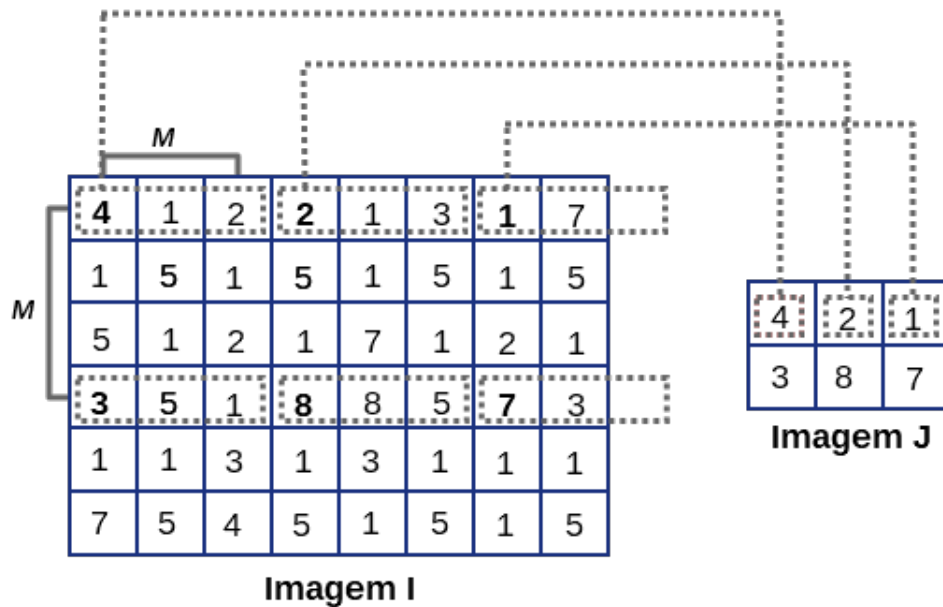


Figura 12 – Exemplo do processo de redimensionamento de imagem por decimação com $M = 3$.

A Figura 13 apresenta exemplos de duas imagens usadas neste trabalho redimensionadas com a técnica de decimação.

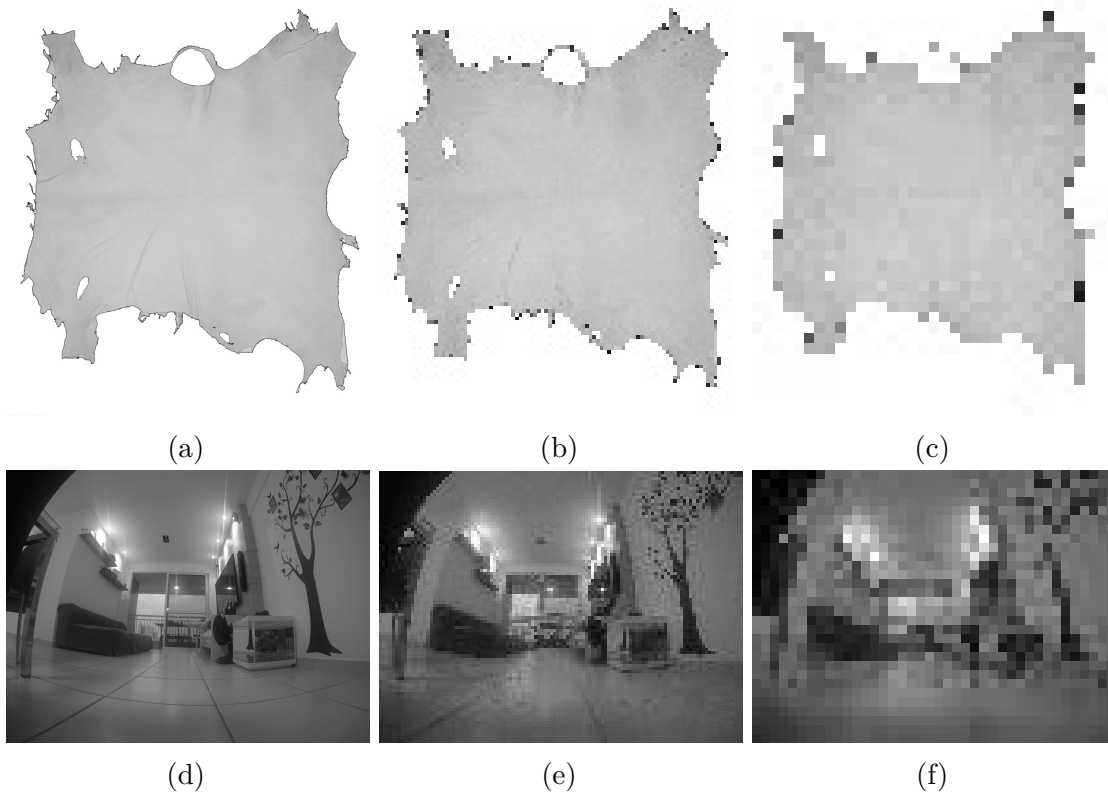


Figura 13 – Exemplos de imagens redimensionada. Da esquerda para a direita as imagens são redimensionadas com $M = \{1, 35, 100\}$. A primeira linha corresponde ao processo com uma imagem da classe 7 presente na base de dados BD1. A segunda linha utiliza uma imagem da classe 7 presente na base de dados BD2.

4.3 Segmentação das Imagens

Segundo Gonzalez e Woods (2010), o processo de segmentação de imagens é uma tarefa que subdivide um imagem em regiões ou objetos que a compõem. O nível de detalhe em que a subdivisão é realizada depende do problema a ser resolvido. A segmentação deve parar quando os objetos ou as regiões de interesse (*Region of Interest* - ROI) de uma aplicação forem detectadas. Por fim, a precisão deste processo pode determinar o sucesso ou o fracasso final dos procedimentos de análise computadorizada.

Após serem convertidas para níveis de cinza com resolução de 8 *bits*, as imagens da base de dados BD1 passaram por um processo de segmentação para extrair somente a área que corresponde ao couro, caracterizando a ROI da aplicação. A Figura 14 mostra as principais etapas desse processo em uma amostra de peça de couro caprino.

O operador Canny proposto por Canny (1986), foi usado com finalidade de melhorar o destaque das bordas dos objetos na imagem. A ideia da filtragem é obter, na imagem, os contornos possibilitando a identificação e segmentação do couro na imagem. Após o processo de filtragem com o operador Canny, obtemos uma imagem, na qual as arestas



Figura 14 – Processo de extração da área do couro.

dos objetos presente na imagem original se destacam. A partir desta imagem, O último passo do sistema é a filtragem dos contornos encontrados na etapa anterior removendo a região da imagem que não faz parte do couro. Assim, toda região dentro do contorno é considerada couro. O resultado final pode ser verificado na Figura 14c. Observa-se que todo o formato da área do couro na imagem original foi mantido na ROI.

Para as imagens da base de dados BD2, não foi necessário o uso de técnicas de segmentação. Ou seja, a imagem completa foi considerada a ROI da aplicação, visto que, toda a representação da cena do ambiente na imagem é essencial para o bom desempenho do reconhecimento e localização do robô móvel.

4.4 Extração, seleção e redução de atributos

A extração de características é uma tarefa essencial do processo de classificação de imagens. As imagens são convertidas em vetores de características a serem utilizados no treinamento dos diversos classificadores de padrões. As características mais usadas para descrever uma imagem de modo sucinto são baseadas principalmente em distribuições de intensidades (cores), textura e forma (SILVIA, 2007).

De acordo com Castañon (2003), extrair atributos mais importantes em uma imagem pode evidenciar as diferenças e similaridades entre os objetos. O principal objetivo da extração de características é realizar uma combinação entre o conjunto de informações fornecidas, criando um espaço de atributos que melhor representa sua discriminabilidade.

A textura pode ser reconhecida e diferenciada pelo olhar humano através de características de suavidade, rugosidade e regularidade (GONZALEZ; WOODS, 2010). A análise de textura é aplicada em várias áreas, como robótica, análise de imagens médicas, pesquisa de imagem e interpretação visual de imagens a partir de sensoriamento remoto (MARINHO et al., 2017).

A partir da ROI de uma imagem, as características são extraídas para serem utilizadas na classificação através de técnicas de aprendizagem de máquina. Neste trabalho foram usados dois descritores baseados em texturas para a extração de características. Um que usa uma técnica de extração de características de imagens digitais a partir de uma matriz de coocorrência de níveis de cinza (Gray Level Co-Occurrence Matrix, GLCM), proposta por Haralick (1979) e descrito na Seção 2.2.1 e o outro baseado em padrões binários locais (Local Binary Patterns, LBP), formulado por Ojala, Pietikainen e Harwood (1994) e apresentado na Seção 2.2.2. Todos os processos de extração de características foram implementados na estação de trabalho ET1.

Como foi apresentado na Seção 2.2, a GLCM disponibiliza um conjunto de 24 medidas estatísticas, que podem ser extraídas de uma única imagem. A Tabela I apresenta a relação de todas essas medidas, que foram adotados como atributos neste trabalho. Inicialmente, todos os 24 atributos foram extraídos de todas as 350 imagens de peças de couro da base de dados BD1, resultando em uma matriz com 8400 medidas estatísticas. Do mesmo modo, uma matriz de 14400 medidas estatísticas foram obtidas com as 600 imagens da base de dados BD2.

Tabela 4 – Relação das 24 medidas estatísticas da GLCM.

Fonte	#	Atributos
Haralick et al. (1973)	14	ASM, Contraste, Correlação, Variância, IDM, Média da Soma I, Variância da Soma, Entropia da Soma, Entropia, Variância da Diferença, Entropia da Diferença, IMC I, IMC II, MCC.
Soh and Tsatsoulis (1999)	6	Homogeneidade, Autocorrelação, Dissimilaridade, Tons de agrupamento, Proeminência do agrupamento, Probabilidade Máxima.
Wang et al. (2010)	4	Média da soma II, Tendência de Agrupamento, Média da diferença, Inércia.
-	24	-

Em relação ao LBP, inicialmente foi utilizada sua implementação original (OJALA; PIETIKAINEN; HARWOOD, 1994) com vizinhança de 8 *pixels*, totalizando $2^8 = 256$ padrões binários, os quais foram usados como características extraídas de uma imagem. Sendo assim, para as bases de dados BD1 e BD2, foram utilizadas, respectivamente, matrizes com 89600 e 153600 informações de características. Os elementos do vetor de características do LBP foram numerados de 1 a 256, onde 1 equivale à quantidade de valores em decimal formados pelo padrão binário 00000000_2 e 256 pelo padrão binário 11111111_2 .

Tendo em vista as possíveis restrições de recursos computacionais em plataformas embarcadas, a utilização de todos os atributos não parece uma boa solução, mesmo que apresentem um bom desempenho no reconhecimento de padrões. Assim, duas abordagens foram aplicadas na redução da quantidade de atributos:

- Seleção de atributos baseada em correlação (CFS).
- Estudo da complexidade na extração dos atributos.

4.4.1 Seleção de atributos baseada em correlação (CFS)

O CFS, proposto por Hall (1998), é um algoritmo que avalia o valor de um subconjunto de atributos considerando a capacidade preditiva individual de cada atributo juntamente com o grau de redundância entre os mesmos. Neste trabalho o CFS foi utilizado para selecionar quais os atributos que mais contribuem na predição correta do classificador com a expectativa de reduzir o vetor de características usado no processo de aprendizagem do classificador SVM. Vale ressaltar que o conjunto resultante contendo os atributos selecionados pelo CFS é inerente ao problema aplicado. Ou seja, o CFS pode selecionar atributos diferentes em aplicações distintas.

A Figura 15 exemplifica o funcionamento do algoritmo CFS. A partir de uma matriz de dados, composta pelos vetores de características associados à classe que cada imagem pertence, o CFS retorna um vetor contendo um subconjunto de atributos que apresenta um bom grau de distinção entre as classes. O tamanho do vetor resultante, dependendo da correlação dos dados, pode ser menor ou igual ao tamanho do vetor de características extraído de uma imagem.

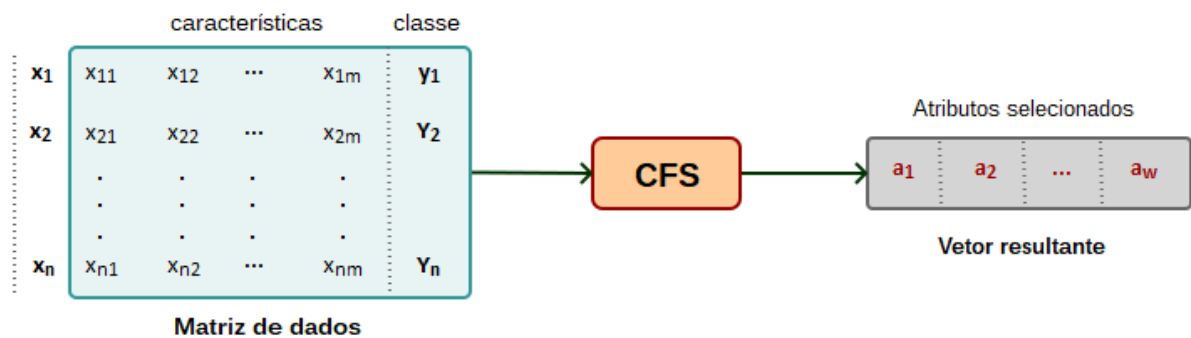


Figura 15 – Ilustração da utilização do algoritmo CFS.

Dentre os 24 atributos da GLCM, o CFS selecionou somente 7 atributos usando a base de dados BD1 e 17 atributos com a base de dados BD2. Já no LBP, dentre os 256 atributos, foram obtidos vetores com tamanhos de 13 e 26 atributos para as bases BD1

e BD2, respectivamente. A Tabela 5 apresenta os conjuntos de atributos resultantes do algoritmo CFS para cada aplicação adotada nesta dissertação.

Tabela 5 – Conjuntos de características resultantes do algoritmo CFS por aplicação.

Base de Dados	Extrator de Características			
	GLCM	#	LBP	#
BD1 (Classificação de Couro)	ASM, Correlação, Variância da diferença, MCC, Probabilidade máxima, Proeminência do agrupamento, Média da diferença	7	2, 3, 9, 11, 13, 17, 29, 36, 115, 131, 137, 239, 249	13
BD2 (Localização do Robô)	ASM, Contraste, Correlação, Variância, Média da soma I, Entropia da soma, Entropia, Variância da diferença, Entropia da diferença, IMC I, IMC II, MCC, Probabilidade máxima, Tons de agrupamento, Proeminência do agrupamento, Dissimilaridade, Média da diferença	17	1, 3, 4, 7, 8, 11, 13, 15, 16, 17, 22, 23, 25, 27, 29, 30, 31, 32, 34, 35, 41, 45, 48, 49, 52, 57	26

Além do uso do CFS, considerando o objetivo de minimizar os custos do sistema embarcado alvo, uma pergunta se faz importante: qual o impacto do cálculo de cada atributo nos custos de desempenho e memória? Dessa forma, foi realizado um estudo de complexidade do cálculo de todos os 24 descritores GLCM, e outro estudo avaliando a complexidade da implementação do LBP.

4.4.2 Estudo da complexidade na extração dos atributos

A ordem de crescimento do tempo de execução de um algoritmo, nos permite comparar o desempenho relativo de algoritmos alternativos Cormen et al. (2009).

O estudo da complexidade se preocupa com a maneira como o tempo de execução de um algoritmo aumenta com o tamanho da entrada n . As notações que geralmente são usadas para descrever o tempo de execução de um algoritmo são chamadas de notações assintóticas. Em geral, um algoritmo que é assintoticamente mais eficiente será a melhor escolha para todas as entradas, exceto as muito pequenas (CORMEN et al., 2009).

Complexidade GLCM

Neste trabalho foi utilizada a notação assintótica Θ para descrever os tempos de execução dos algoritmos de cálculo de todos os 24 atributos GLCM (Tabela 4) inspecionando os desempenhos no pior caso.

Neste estudo, o tamanho da entrada n do algoritmo para calcular os atributos é o valor da dimensão da Matriz de Coocorrência MC. Sendo que é preciso percorrer toda a matriz para extrair um atributo.

Na Tabela 6 são apresentados todos os 24 atributos de acordo com a complexidade do tempo de execução, baseado nas equações de Haralick, Shanmugam e Dinstein (1973), Soh e Tsatsoulis (1999) e Wang et al. (2010). Analisando todos valores para os cálculos dos atributos, a menor complexidade de tempo de execução encontrada foi $\Theta(n^2)$.

Tabela 6 – Relação dos atributos agrupados por custo de complexidade temporal

Custo Temporal	Atributos
$\Theta(n^2)$	ASM, Contraste, IDM, Entropia, Homogeneidade, Média da soma II, Probabilidade máxima, Dissimilaridade, Média da diferença, Autocorrelação.
$\Theta(n^2 + n)$	Correlação, Inércia.
$\Theta(n^2 + 2n)$	Média da soma I, Entropia da soma, Variância da diferença, Entropia da diferença.
$\Theta(n^2 + 4n - 2)$	Variância da Soma.
$\Theta(2n^2)$	Variância, IMC I, IMC II, MCC, Tendência de agrupamento, Tons de agrupamento, Proeminência do agrupamento.

Verifica-se que alguns atributos selecionados pelo CFS, por exemplo o Proeminência do agrupamento (*Cluster Prominence*), apresentaram um tempo de computação $\Theta(2n^2)$, superior a $\Theta(n^2)$, o que pode comprometer a implementação em uma plataforma de recursos limitados.

Cabe ainda destacar que, embora no grupo $\Theta(n^2)$ haja 10 atributos, eles são todos calculados com apenas dois laços aninhados. Ou seja, percorrendo a MC somente uma vez.

Com essa mesma abordagem, foi feito um novo estudo de complexidade baseado no consumo de memória. Novamente, n representa a dimensão da matriz de coocorrência e cada elemento da matriz consiste em um valor de ponto flutuante. A Tabela 7 apresenta o estudo de complexidade espacial na obtenção de cada atributo, agrupados de forma crescente.

Tabela 7 – Relação dos atributos agrupados por custo de complexidade espacial.

Custo Espacial	Atributos
$\Theta(1)$	ASM, Contraste, Variância, IDM, Entropia, Homogeneidade, Média da Soma II, Probabilidade máxima, Tendência de agrupamento, Tons de agrupamento, Proeminência do agrupamento, Dissimilaridade, Média da diferença, Autocorrelação, Inércia .
$\Theta(n)$	Variância da diferença, Entropia da diferença.
$\Theta(2n - 1)$	Média da Soma I, Entropia da Soma.
$\Theta(2n)$	Correlação, Variância da Soma , IMC I, IMC II, MCC.

Os atributos do grupo que apresentam uma complexidade espacial $\Theta(1)$, consomem apenas um espaço de memória equivalente ao tamanho da matriz de coocorrência.

Constata-se que o grupo que contém os atributos com complexidade temporal $\Theta(n^2)$ está incluído no conjunto de atributos com complexidade espacial $\Theta(1)$. Assim, os 10 atributos com menor tempo de computação também apresentam a menor complexidade espacial.

Complexidade LBP

Nesta pesquisa, a implementação do LBP foi baseada em uma versão otimizada para aplicações que necessitam da extração de características em tempo real apresentado em Mäenpää, Turtinen e Pietikäinen (2003). Neste caso o LBP foi implementado sem saltos condicionais. Segundo o autor, um problema aparente com o operador LBP é que cada valor de *pixel* em uma vizinhança deve ser comparado com o valor do centro. Com P *pixels* vizinhos, isso causa, em média, $P/2$ *delays* na execução das instruções por *pixel*, já que o resultado da comparação não pode ser previsto.

O custo do algoritmo da versão otimizada do LBP pode ser representado por $\Theta(L.C)$ para imagens com dimensões $L \times C$, visto que para encontrar os padrões binários será necessário percorrer toda a matriz (imagem). O custo de computação para formar o padrão binário é constante, já que no seu cálculo apresenta apenas operações aritméticas simples e deslocamento de *bits*.

Embora o algoritmo retorne um histograma representado por um vetor de 256 padrões binários diferentes, neste trabalho foram utilizadas duas variações do LBP com objetivo de diminuir o tempo de computação e a representatividade do vetor de características na memória. Sendo assim, utilizou-se uma versão do LBP com padrões binários uniformes que resultam em um vetor de 59 padrões (PIETIKÄINEN et al., 2011b). Também foi abordado uma extensão do LBP que utiliza apenas $P = 4$ *pixels* vizinhos ao invés de 8 *pixels* na sua versão original. Para o LBP com $P = 4$ serão encontrados padrões binários locais de 4 *bits* resultando em um vetor de características com apenas 16 atributos. A Figura 16 exemplifica o cálculo de um padrão binário local de 4 *bits*.

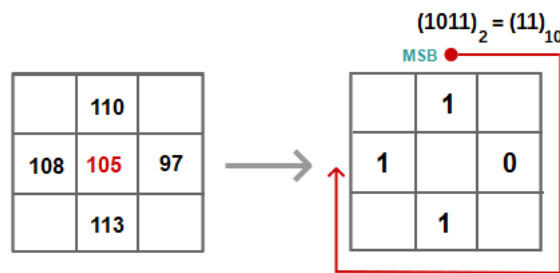


Figura 16 – Exemplo do cálculo do LBP considerando 4 *pixels* vizinhos.

Neste trabalho, com o objetivo de investigar e encontrar quais os melhores atributos, visando a um baixo consumo de recursos computacionais em uma plataforma limitada, vários conjuntos de atributos foram avaliados em conjunto com o classificador SVM. A Tabela 8 apresenta todos os conjuntos de atributos utilizados nesta dissertação. Cada conjunto foi nomeado e avaliado, sendo os resultados apresentados no Capítulo 5.

Tabela 8 – Conjuntos de atributos utilizados na dissertação.

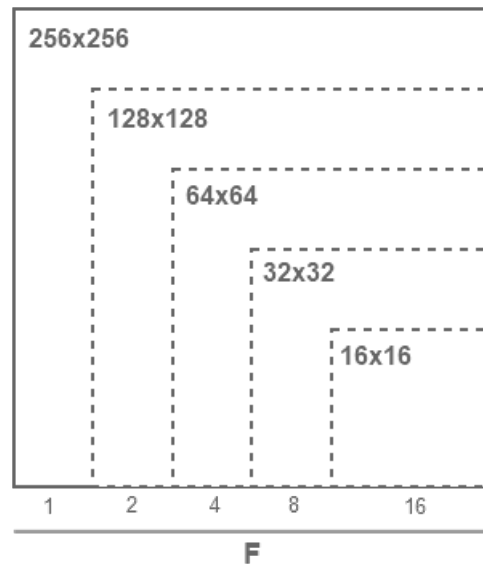
Base	Extrator	Conjunto	Atributos	#	Descrição
BD1 (Classificação de Couro)	GLCM	A1	Todos da Tabela 4.	24	24 atributos GLCM.
		A2	ASM, Correlação, Variância da diferença, MCC, Probabilidade máxima, Proeminência do agrupamento, Média da diferença.	7	Uso do CFS em A1 .
		A3	ASM, Contraste, IDM, Entropia, Homogeneidade, Média da soma II, Probabilidade máxima, Dissimilaridade, Média da diferença, Autocorrelação.	10	Estudo da complexidade.
		A4	ASM, Contraste, Correlação, Homogeneidade.	4	Os mesmos utilizados em SANTOS FILHO (2013).
	LBP	B1	Todos.	256	LBP original
		B2	Todos os padrões uniformes.	59	58 (uniformes) + 1 (não uniformes).
		B3	2, 3, 9, 11, 13, 17, 29, 36, 115, 131, 137, 239, 249.	13	Uso do CFS em B1 .
		B4	Todos.	16	Uso do LBP de 4 <i>bits</i> .
BD2 (Localização do Robô)	GLCM	C1	A1	24	Os mesmos atributos de A1 .
		C2	ASM, Contraste, Correlação, Variância, Média da soma I, Entropia da soma, Entropia, Variância da diferença, Entropia da diferença, IMC I, IMC II, MCC, Probabilidade máxima, Tons de agrupamento, Proeminência do agrupamento, Dissimilaridade, Média da diferença.	17	Uso do CFS em C1 .
		C3	A3	10	Os mesmos atributos de A3 .
		C4	ASM, Contraste, Correlação, Variância, IDM, Média da soma I, Variância da soma, Entropia da soma, Entropia, Variância da diferença, Entropia da diferença, IMC I, IMC II, MCC.	14	Os mesmos utilizados em Marinho (2017).
	LBP	D1	B1.	256	Os mesmos atributos de B1 .
		D2	B2.	59	Os mesmos atributos de B2 .
		D3	1, 2, 3, 4, 7, 8, 11, 16, 17, 18, 25, 29, 31, 32, 34, 35, 41, 45, 47, 49, 57, 61, 62, 63, 92, 96, 97, 109, 113, 120, 121, 125, 127, 130, 131, 132, 135, 136, 143, 144, 161, 164, 172, 181, 188, 193, 194, 196, 200, 201, 202, 225, 226, 228, 230, 233, 239, 241, 242, 244, 249.	61	Uso do CFS em D1 .
		D4	B4.	16	Os mesmos atributos de B4 .

4.5 Redução da Matriz de Coocorrência (MC)

Uma Matriz de Coocorrência MC considera $N_g = 256$ níveis de cinza encontrados nas imagens com resolução de 8 *bits*. Porém, sabendo que a dimensão desta matriz corresponde à quantidade de níveis de cinza da imagem, sua representatividade na memória pode se tornar elevada.

Com o intuito de diminuir o espaço de armazenamento ocupado na memória pela MC no ambiente embarcado a matriz MC foi reduzida a partir de um fator F , o qual representa o valor da fração de tons de cinza da imagem. Por exemplo, para $F = 1$ e $F = 2$ a dimensão da MC, bem como a quantidade de níveis de cinza considerados na imagem, irá resultar, respectivamente, em 256×256 e 128×128 , conforme Figura 17.

Figura 17 – Ilustração do redimensionamento da matriz de coocorrência (CM)



4.6 Classificação e Avaliação

Esta seção apresenta os procedimentos realizados no processo de classificação e avaliação do classificador utilizado neste trabalho. Também são descritas as métricas adotadas na avaliação dos resultados obtidos nos testes. Todas as métricas são baseadas em uma matriz de classificação (ou confusão) do modelo.

O classificador utilizado neste trabalho foi o SVM. Segundo ROCHA NETO e Barreto (2009) as SVM são classificadores que se baseiam na teoria de aprendizado estatístico que leva em consideração a minimização do erro estrutural e não apenas a minimização do erro quadrático médio calculado para os vetores de treinamento.

4.6.1 Treinamento

Geralmente, para estimar a taxa de acerto e taxa de erro obtidas por um classificador em aplicações de reconhecimento de padrões, é comum a divisão da base de dados em treinamento e teste. Esta divisão é chamada *Holdout* (PERLIN, 2015), e tem como objetivo avaliar a generalização de um modelo ou classificador, no caso o SVM. O conjunto de treinamento tem como objetivo oferecer exemplos para o aprendizado do algoritmo. Já o conjunto de testes é utilizado após o fim do treinamento do classificador, a fim de avaliar se a generalização desejada foi atingida.

Para o treinamento do classificador SVM, a quantidade de imagens de cada classe foi dividida aleatoriamente com as seguintes proporções: 74% das imagens para o conjunto de treinamento e 26% para o conjunto de teste. O *kernel* usado foi o RBF com um parâmetro manual de $\sigma = 2$. Para o classificador foram executadas 50 realizações ou rodadas independentes de treinamento e teste.

4.6.2 Testes

Na realização dos testes do classificador, a fim de verificar o seu desempenho de generalização, foram usadas 5 medidas para a avaliação do modelo. Essas cinco métricas foram calculadas a partir de uma matriz de confusão.

4.6.2.1 Métricas de Avaliação

Com o intuito de analisar o desempenho do classificador SVM, cinco métricas de avaliação são empregadas e definidas a seguir: acurácia, sensibilidade, especificidade, valor preditivo positivo e *F1-score*. Essas medidas são obtidas a partir de uma matriz de confusão. Conforme apresentado em Fawcett (2006) e Powers (2011), o cálculo de cada medida é dado da seguinte forma:

* Considera-se k uma classe qualquer para análise.

- Verdadeiro Positivo (VP): corresponde ao total de padrões k que são classificados como k .
- Falso Negativo (FN): corresponde ao total de padrões k que não são classificados como k .
- Falso Positivo (FP): corresponde ao total de padrões diferentes de k que são classificados como k .
- Verdadeiro Negativo (VN): corresponde ao total de padrões diferentes de k que não são classificados como k .

Uma matriz de classificação é criada classificando-se todas as imagens em categorias ou classes, determinando se o valor previsto correspondeu ao valor real. Todas as imagens

em cada classe são contabilizadas e os totais são exibidos na matriz. As linhas na matriz representam os rótulos reais das classes, sendo que as colunas representam as classes previstas. Essa matriz é uma ferramenta padrão para avaliação de modelos estatísticos e é muitas vezes chamada de matriz de confusão.

As Tabelas 9 e 10 apresentam uma matriz de confusão para um problema de 7 classes. Com objetivo de facilitar a compreensão das métricas apresentadas, foram destacadas, respectivamente, as classes 1 e 4 em negrito, uma vez que o VP, FN, FP e VN são relativos às classes analisadas.

Tabela 9 – Exemplo de uma matriz de confusão para um problema de 7 classes analisando a classe 1.

Classe	1	2	3	4	5	6	7
1	VP	FN	FN	FN	FN	FN	FN
2	FP	VN	VN	VN	VN	VN	VN
3	FP	VN	VN	VN	VN	VN	VN
4	FP	VN	VN	VN	VN	VN	VN
5	FP	VN	VN	VN	VN	VN	VN
6	FP	VN	VN	VN	VN	VN	VN
7	FP	VN	VN	VN	VN	VN	VN

Tabela 10 – Exemplo de uma matriz de confusão para um problema de 7 classes analisando a classe 4.

Classe	1	2	3	4	5	6	7
1	VN	VN	VN	FP	VN	VN	VN
2	VN	VN	VN	FP	VN	VN	VN
3	VN	VN	VN	FP	VN	VN	VN
4	FN	FN	FN	VP	FN	FN	FN
5	VN	VN	VN	FP	VN	VN	VN
6	VN	VN	VN	FP	VN	VN	VN
7	VN	VN	VN	FP	VN	VN	VN

A acurácia (Ac) é definida como a relação entre o número de padrões corretamente classificados e o número total de padrões para teste:

$$Ac = \frac{\sum \text{padrões corretamente classificados}}{\sum \text{padrões}} = \frac{\sum VP + \sum VN}{\sum \text{padrões}}. \quad (4.1)$$

Sensibilidade (Se) é a relação entre o número de padrões corretamente classificados de uma determinada classe e o número total de padrões disponível para essa classe, incluindo aqueles que foram mal classificados:

$$Se = \frac{\sum VP}{\sum VP + \sum FN} \quad (4.2)$$

A especificidade (Es) mede a proporção de negativos que estão corretamente classificados:

$$Es = \frac{\sum VN}{\sum VN + \sum FP} \quad (4.3)$$

O valor preditivo positivo (VPP) é a relação entre os padrões corretamente classificados de uma classe específica e todos os padrões classificados pertencentes a essa classe, isto é, mede a proporção de verdadeiros positivos em relação à todas as predições positivas:

$$VPP = \frac{\sum VP}{\sum VP + \sum FP} \quad (4.4)$$

O $F1$ -Score é a média harmônica ponderada entre o VPP a sensibilidade. Um valor de $F1$ -Score atinge o seu melhor valor em 1 e pior resultado em 0.

$$F1\text{-Score} = 2 \cdot \frac{VPP \cdot Se}{VPP + Se} \quad (4.5)$$

Formalmente, para o cálculo das métricas analisando uma classe arbitrária k a partir de uma matriz de confusão A resultante de um problema de n classes, primeiramente, calcula-se o somatório dos valores de VP , FN , FP e VN da seguinte forma:

- $\sum VP = A(k, k)$;
- $\sum FN = \sum_{i=1}^n A(k, i) - A(k, k)$;
- $\sum FP = \sum_{i=1}^n A(i, k) - A(k, k)$;
- $\sum VN = \sum \text{padrões} - (\sum VP + \sum FN + \sum FP)$.

Com base na matriz de confusão representada na Tabela 11 e na formulação descrita acima, o cálculo da acurácia Ac para o reconhecimento de padrões considerando a classe 1, é realizado da seguinte maneira:

- $\sum VP = 12$;
- $\sum FN = 13 - 12 = 1$;
- $\sum FP = 15 - 12 = 3$;
- $\sum VN = 91 - 16 = 75$;
- $Ac = \frac{12+75}{91} = 0,956$ (95,6%).

Tabela 11 – Exemplo de uma matriz de confusão com valores reais.

Classe	1	2	3	4	5	6	7
1	12	0	0	0	0	0	1
2	1	9	0	2	0	0	1
3	0	0	10	2	0	0	1
4	0	1	0	11	1	0	0
5	0	0	0	2	10	0	1
6	2	0	0	0	0	10	1
7	0	0	0	1	1	1	10

4.7 Conclusão

O capítulo apresentou as aplicações usadas como casos de estudo nesta dissertação, assim como suas bases de dados. A técnica utilizada no processo de redimensionamento das imagens foi explorada. Foram também apresentadas duas abordagens para seleção e redução dos atributos. Uma baseada na correlação entre os atributos, e a outra a partir da complexidade na extração dos atributos. Além disso, uma abordagem para reduzir o tamanho da matriz GLCM foi apresentada. Em seguida uma relação contendo todos os conjuntos de atributos que foram utilizados nos experimentos foi apresentada. Finalmente, o processo de classificação (treino e teste), assim como as métricas de avaliação foram discutidas.

No próximo capítulo serão apresentados os resultados obtidos utilizando as abordagens apresentadas com objetivo de identificar uma solução adequada para embarcar os processos de reconhecimento de padrões de cada aplicação em um plataforma alvo.

5 Resultados e Discussões

Neste capítulo são apresentados os resultados e os experimentos realizados neste trabalho com base nos procedimentos demonstrados no Capítulo 4. Para cada uma das variações propostas, o desempenho do processo de classificação foi avaliado em uma estação de trabalho. O capítulo finaliza apresentando os resultados de tempo de computação e consumo de memória na plataforma embarcada para cada aplicação adotada. Em cada grupo os resultados por extratores de atributos são explorados separadamente.

Os resultados referentes à aplicação AP1 (classificação de couro) são provenientes de uma base de dados contendo 350 imagens com dimensões 3456×4608 . Essas imagens são categorizadas em 7 classes, onde cada classe apresenta 50 amostras de peças de couro caprino.

Já os resultados referentes à aplicação AP2 (localização e navegação de um robô móvel) são obtidos de uma base de dados contendo 600 imagens com dimensões 4000×3000 . As imagens foram categorizadas em 15 classes, onde cada classe apresenta 40 amostras de ambientes internos.

Também foram avaliados os desempenhos do classificador SVM combinado com os extratores GLCM e LBP para cada aplicação utilizando as imagens redimensionadas.

Os experimentos foram computados nas estações de trabalho ET1 e ET2. Os testes no ambiente embarcado foram realizados na plataforma embarcada Raspberry Pi.

5.1 Desempenho do Classificador nas Estações de Trabalho

Na estação de trabalho foram feitos experimentos para avaliar: o custo de computação no processo de classificação para cada conjunto de atributos, antecipando o impacto na plataforma embarcada, o desempenho do classificador para imagens com diferentes resoluções, e por último o consumo memória para cada extrator de atributos.

O classificador SVM foi treinado nas estações de trabalho ET1 e ET2, verificando-se a partir de experimentos, que apresentaram resultados iguais. A plataforma embarcada alvo utiliza OpenCV e a estação ET2 serve como validação intermediária. O *kernel* usado no SVM foi o RBF com um parâmetro manual de $\sigma = 2$. No treinamento foram utilizadas 50 realizações ou rodadas.

5.1.1 Custo de Computação

As medições do uso do tempo foram realizadas na estação de trabalho ET1. O seguinte procedimento foi adotado: foram realizadas seis medidas, descartou-se a primeira e foi calculada a média das cinco restantes. A primeira medida foi descartada na tentativa de eliminar a interferência do tempo de carregamento das bibliotecas do Matlab no custo de computação. Nas medições dos tempos de computação apenas a ferramenta Matlab estava rodando na estação de trabalho ET1 e todos os outros processos foram postos em *standby*.

Em todos os experimentos desta seção foram utilizadas imagens das aplicações no seu tamanho original. Ou seja, imagens com resoluções de 3456×4608 *pixels* da AP1 e 4000×3000 *pixels* da AP2.

5.1.1.1 GLCM

Com a ferramenta Matlab foram medidos os tempos de computação nas etapas do processo de classificação utilizando o SVM combinado com o extrator GLCM. Em cada etapa foi calculado uma média entre cinco medidas de tempo. Neste experimento foram utilizadas as seguintes etapas: carregamento da imagem, criação e normalização da matriz MC, extração de características e classificação.

A Tabela 12 apresenta as medidas de tempos totais de processamento nas aplicações AP1 e AP2 usando o extrator GLCM. Mesmo obtendo maior tempo, a utilização do conjunto A3 na AP1 se destaca em relação ao A4, já que o A3 obteve desempenhos de acurácia melhores. Além disso, foi verificado que o uso dos 10 atributos GLCM com menor complexidade temporal (A3) obteve menor tempo, comparado com a utilização dos 7 atributos selecionados pelo algoritmo CFS.

Na aplicação AP2, com o uso do conjunto C3 (A3) foi obtido um tempo de computação de 2,3s. Esse resultado apresenta um custo de computação inferior aos tempos obtidos utilizando os atributos C1, C2 e C4.

Tabela 12 – Tempos de computação (em segundos) na estação de trabalho para as aplicações AP1 e AP2. Utilização do extrator GLCM com imagens no tamanho original.

Aplicação	Atributos	#	Tempo (s)	Ac(%)
AP1	A1	24	4,3927	$90,49 \pm 1,26$
	A2 : CFS(A1)	7	4,0805	$91,10 \pm 1,06$
	A3	10	3,3295	$90,78 \pm 0,98$
	A4	4	1,5295	$89,44 \pm 1,09$
AP2	C1	24	3,3626	$99,76 \pm 0,14$
	C2 : CFS(C1)	17	3,3395	$99,74 \pm 0,14$
	C3	10	2,3018	$99,68 \pm 0,13$
	C4	14	3,3301	$99,42 \pm 0,25$

5.1.1.2 LBP

Do mesmo modo, foram obtidos os tempos de computação nas etapas do processo de classificação utilizando o SVM combinado com o extrator LBP. Também foi calculado uma média entre cinco medidas de tempo. Neste caso foram utilizadas as seguintes etapas: carregamento da imagem, extração de características e classificação. A Tabela 13, apresenta as medidas de tempos totais de processamento nas aplicações AP1 e AP2, usando o extrator LBP.

Observando a Tabela 13, verifica-se que o tempo de computação com os atributos B2 e B3 da aplicação AP1, mesmo com uma quantidade menor de atributos, foram maiores em relação ao B1 de 256 atributos. Isso ocorre porque na implementação os 256 atributos ainda precisam ser calculados e ainda é adicionado um custo de computação da escolha dos 59 padrões binários uniformes e na escolha dos atributos resultantes do algoritmo CFS. Esse mesmo comportamento é apresentado nos resultados com os atributos D2 e D3 na aplicação AP2. Por outro lado, o custo de computação com os atributos de B4 foi aproximadamente de 5s na aplicação AP1. Na aplicação AP2 o custo de computação com os atributos de D4 (os mesmo de B4) foi de 3,9s.

Tabela 13 – Tempos de computação (em segundos) na estação de trabalho para as aplicações AP1 e AP2. Utilização do extrator LBP com imagens no tamanho original.

Aplicação	Atributos	#	Tempo (s)	Acurácia (%)
AP1	B1	256	41,8256	78,15 $\pm$ 0,82
	B2	59	90,3543	87,93 $\pm$ 1,38
	B3 : CFS(B1)	13	52,9260	92,90 $\pm$ 0,97
	B4	16	5,1234	89,75 $\pm$ 0,98
AP2	D1	256	45,2834	99,42 $\pm$ 0,18
	D2	59	67,5155	99,86 $\pm$ 0,11
	D3 : CFS(D1)	61	65,8253	99,85 $\pm$ 0,10
	D4	16	3,9536	99,89 $\pm$ 0,10

5.1.1.3 Discussão parcial dos resultados

Os atributos GLCM selecionados a partir do estudo de complexidade apresentado neste trabalho, A3 e C3, obtiveram baixos custos de computação em ambas as aplicações. Verificou-se também que o extrator LBP, no geral, apresentou tempos de computação superiores aos do GLCM.

No resultados, os custos computacionais utilizando todos os 256 atributos do LBP, obtiveram tempos de 41,8s e 45,2s, respectivamente, nas aplicações AP1 e AP2. Sendo assim, o uso de todos os 256 padrões binários pode ser considerado inviável em aplicações embarcadas onde o tempo de resposta do projeto é limitado. Isso utilizando as imagens no seu tamanho original. Contudo, a utilização de outras extensões do LBP original pode ser uma boa solução, como é o caso dos 16 atributos de B4 e D4. Esses, oriundos dos padrões binários formados a partir de apenas 4 *pixels* vizinhos.

A nível de comparação, a Tabela 14 apresenta os tempos de computação das melhores soluções GLCM e LBP para cada aplicação. Como esperado, os tempos de computação na AP1 foram maiores que na AP2. Isso se deve as imagens da AP1 serem maiores que as imagens da aplicação AP2. Vale lembrar A3=C3 e B4=D4, e que neste experimento as imagens estão no seu tamanho original.

Tabela 14 – Tempos de computação (em segundos) das melhores soluções GLCM e LBP na estação de trabalho para as aplicações AP1 e AP2. Utilização das imagens no tamanho original.

Aplicação	Extrator	Atributos	#	Tempo (s)	Ac(%)
AP1	GLCM	A3	10	3,3295	$90,78 \pm 0,98$
	LBP	B4	16	5,1234	$89,75 \pm 0,98$
AP2	GLCM	C3	10	2,3018	$99,68 \pm 0,13$
	LBP	D4	16	3,9536	$99,89 \pm 0,10$

Este estudo possibilita antecipar o impacto da utilização dos conjuntos de atributos para cada aplicação na plataforma embarcada. Neste trabalho, a partir dos resultados apresentados, foram escolhidos os conjuntos de atributos A3, B4, C3 e D4 para serem experimentados em um ambiente embarcado.

5.1.2 Resolução das imagens

Para o redimensionamento das imagens foram utilizados $1 \leq M \leq 1000$. Considerando apenas os valores de M que causam variações na dimensão da imagem. Sendo assim, foram usados 163 valores diferentes para M no total. Em resoluções efetivas, as imagens foram reduzidas de (3456×4608) a (3×4) para a aplicação AP1 e (4000×3000) a (4×3) para a AP2.

Na análise de desempenho do classificador com imagens redimensionadas, o classificador SVM foi treinado novamente com o conjunto de imagens de tamanho reduzido.

Em ambas aplicações foi avaliado o desempenho do classificador para cada classe. Neste caso, as escolhas dos valores de M e dos conjuntos de atributos GLCM e LBP foram baseadas no estudo apresentado na Seção 5.1.3.

5.1.2.1 GLCM

Neste trabalho, o extrator de atributos GLCM foi avaliado em cada aplicação com as imagens redimensionadas. O intuito desses experimentos é investigar a influência da resolução das imagens no desempenho de acurácia do classificador SVM combinado com o GLCM.

Resultados na aplicação AP1 (Classificação de Couro)

Na aplicação AP1, foi avaliado o desempenho do classificador SVM combinado com o extrator GLCM usando imagens redimensionadas por M . Neste caso, para os diferentes tamanhos das imagens foram extraídos os atributos GLCM de A1, A2, A3 e A4.

A Figura 18 apresenta a variação no desempenho do classificador SVM com o extrator GLCM na aplicação AP1, conforme varia o valor de M . Para este estudo foram utilizados os conjuntos de atributos A1, A2, A3 e A4.

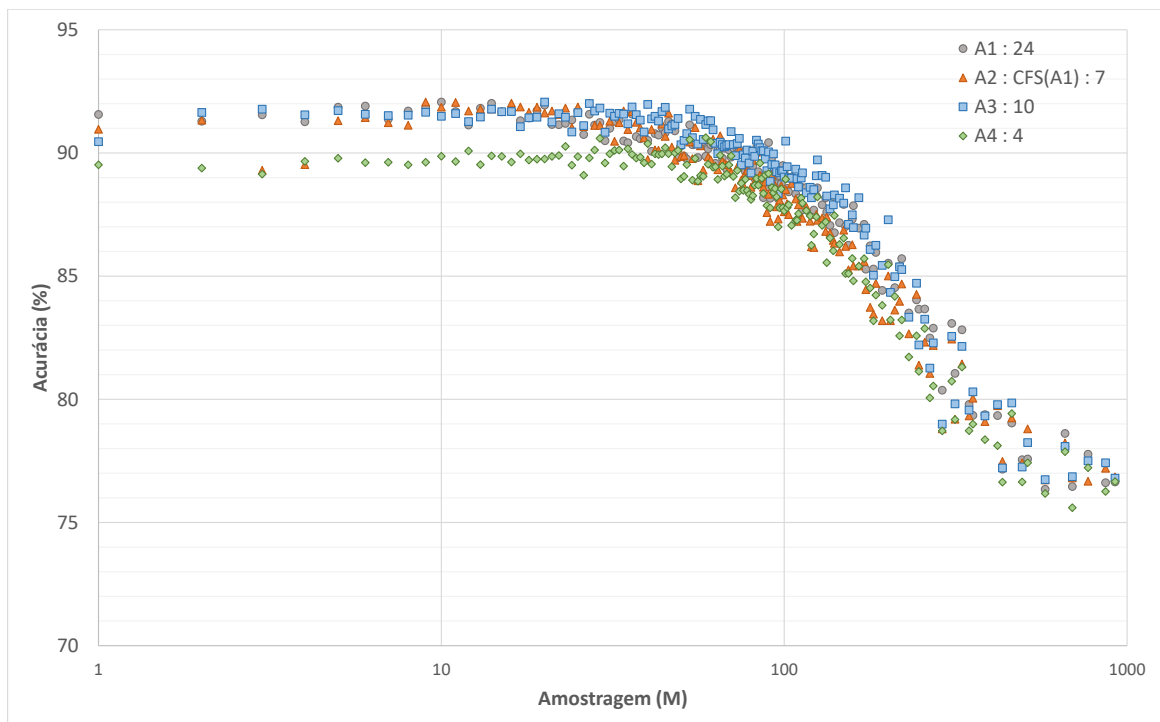


Figura 18 – Desempenho do classificador SVM com o extrator GLCM na aplicação AP1, utilizando imagens redimensionadas.

Ao analisar a Figura 18, verifica-se que, na medida que o tamanho da imagem diminui até $M = 100$ (34×46 pixels), a acurácia cai discretamente, sem variações muito bruscas.

O desempenho com o conjunto A2 (7 atributos) demonstra a eficiência do CFS na escolha de atributos para o reconhecimento de padrões na aplicação de classificação de couro. Embora tenha obtido um bom desempenho para $M \leq 100$ e com apenas 7 atributos, A2 contém atributos com complexidade temporal superior aos atributos de A3.

A utilização dos 10 atributos de A3, que apresentam menor complexidade temporal, produziu taxas de acerto praticamente iguais com a utilização de A1 (24 atributos) e A2 (7 atributos), mas, superior ao desempenho com os atributos de A4 (4 atributos), os mesmos utilizados em SANTOS FILHO (2013).

A Tabela 15 mostra os valores médios e os desvios padrões da acurácia (Ac), especificidade (Es), sensibilidade (Se), vapor preditivo positivo (VPP) e F1-Score obtidos pela combinação do classificador SVM com o extrator GLCM utilizando os atributos de A3 para imagens redimensionadas da aplicação AP1.

Tabela 15 – Desempenho do classificador SVM combinado com o extrator GLCM usando o conjunto de atributos A3 na aplicação AP1.

M	Tamanho	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score(%)
1	3456 × 4608	90,78 ± 0,98	67,71 ± 3,44	94,62 ± 0,57	68,19 ± 3,71	66,78 ± 3,60
2	1728 × 2304	91,51 ± 1,12	70,29 ± 3,91	95,05 ± 0,65	72,36 ± 4,23	69,50 ± 4,12
5	691 × 921	91,46 ± 1,11	70,11 ± 3,90	95,02 ± 0,65	72,31 ± 3,95	69,63 ± 3,98
10	345 × 460	91,31 ± 1,24	69,58 ± 4,35	94,93 ± 0,73	72,08 ± 4,86	68,96 ± 4,53
20	172 × 230	92,03 ± 1,32	72,09 ± 4,64	95,35 ± 0,77	74,85 ± 4,79	71,54 ± 4,82
30	115 × 153	90,93 ± 1,15	68,26 ± 4,02	94,71 ± 0,67	70,40 ± 4,81	67,19 ± 4,36
40	86 × 115	91,95 ± 1,21	71,82 ± 4,22	95,30 ± 0,70	74,34 ± 4,20	71,25 ± 4,29
50	69 × 92	90,44 ± 1,10	66,55 ± 3,84	94,42 ± 0,64	69,66 ± 4,58	65,78 ± 4,22
60	57 × 76	91,22 ± 1,06	69,27 ± 3,71	94,88 ± 0,62	71,77 ± 4,37	68,30 ± 3,97
70	49 × 65	90,54 ± 1,23	66,90 ± 4,30	94,48 ± 0,72	69,64 ± 4,06	66,25 ± 4,33
73	47 × 63	90,33 ± 1,12	66,15 ± 3,92	94,36 ± 0,65	68,64 ± 4,33	65,04 ± 4,31
80	43 × 57	89,41 ± 1,19	62,92 ± 4,16	93,82 ± 0,69	65,59 ± 4,80	61,59 ± 4,42
90	38 × 51	90,04 ± 1,07	65,14 ± 3,75	94,19 ± 0,63	69,17 ± 4,42	64,22 ± 4,10
100	34 × 46	89,12 ± 0,98	61,91 ± 3,43	93,65 ± 0,57	65,29 ± 3,45	61,04 ± 3,40

O desempenho do classificador para cada classe está apresentado na Tabela 16. Para esses resultados utilizou-se do conjunto A3 e imagens redimensionadas com $M = 73$ (47×63). Foi verificado que em valores médios de acurácia, o classificador obteve um melhor reconhecimento para os padrões das classes 5 e 6, nos quais foram acima de 96%.

Tabela 16 – Desempenho do classificador SVM por classe, utilizando o conjunto de atributos A3 e imagens redimensionadas por $M = 73$ (47×63) pixels.

Classe	Métricas				
	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score
1	85,98 ± 02,46	83,85 ± 08,96	86,33 ± 03,08	51,05 ± 05,29	63,16 ± 04,97
2	88,24 ± 02,44	75,38 ± 12,14	90,38 ± 02,91	57,20 ± 06,84	64,54 ± 07,22
3	88,88 ± 02,50	49,23 ± 13,09	95,49 ± 02,04	65,85 ± 13,77	55,19 ± 12,25
4	88,75 ± 02,50	48,46 ± 14,01	95,46 ± 02,17	64,75 ± 13,65	54,46 ± 12,09
5	96,81 ± 01,57	83,08 ± 10,43	99,10 ± 00,98	94,34 ± 05,90	87,90 ± 06,35
6	96,31 ± 01,96	88,92 ± 10,56	97,54 ± 01,66	86,41 ± 08,34	87,18 ± 07,46
7	87,34 ± 02,45	88,92 ± 10,56	97,54 ± 01,66	86,41 ± 08,34	87,18 ± 07,46
Média	90,33 ± 01,12	66,15 ± 03,92	94,36 ± 00,65	68,64 ± 04,33	65,04 ± 04,30

Resultados na aplicação AP2 (Localização do Robô)

Os resultados obtidos no desempenho do classificador SVM combinado com GLCM na aplicação AP2 estão apresentados na Figura 19. Para este estudo foram utilizados os conjunto de atributos C1, C2, C3 e C4 e valores de $1 \leq M \leq 1000$.

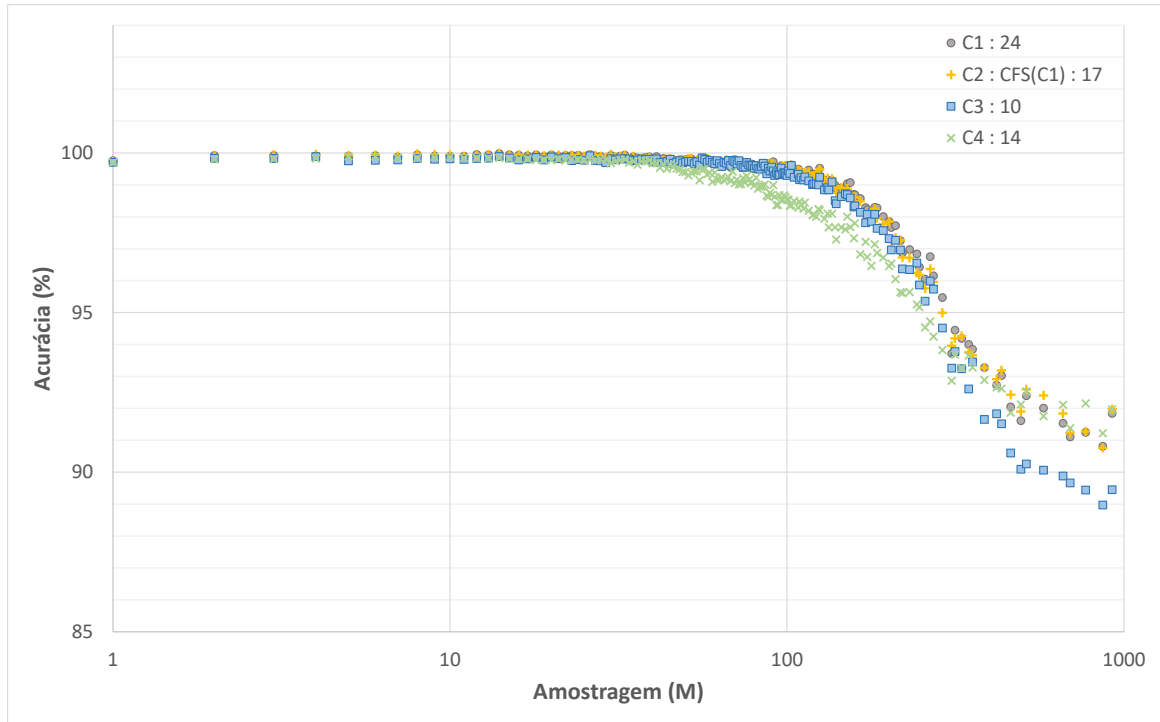


Figura 19 – Desempenho do classificador com GLCM para a aplicação AP2 utilizando imagens redimensionadas.

Analisando a Figura 19, verifica-se que, para valores de $M \leq 100$, o desempenho do classificador para os conjuntos de atributos praticamente foram iguais. A acurácia chegou a alcançar valores médios até 99%.

O CFS também se mostrou eficiente com a GLCM para a aplicação AP2. A utilização dos 10 atributos ($C3=A3$) de menor complexidade temporal, produziu taxas de acerto superior ao desempenho com os atributos de C4 (14 atributos), os mesmos utilizados em Marinho (2017).

A Tabela 17 apresenta os valores médios e os desvios padrões das métricas obtidas pela combinação do classificador SVM com o extrator GLCM utilizando os atributos de C3 para imagens redimensionadas da aplicação AP2.

O desempenho do classificador para cada classe está apresentado na Tabela 18. Para esses resultados utilizou-se do conjunto de atributos C3 e imagens redimensionadas com $M = 38$ (105×78). Foi verificado que em valores médios de todas as métricas, o classificador obteve um excelente desempenho, chegando a alcançar para o reconhecimento de padrões das classes 4 e 13, 100% de acertos.

Tabela 17 – Desempenho do classificador SVM combinado com o extrator GLCM usando o conjunto de atributos C3 na aplicação AP2.

M	Tamanho	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score(%)
1	4000 × 3000	99.42 ± 0.25	95.85 ± 1.76	99.69 ± 0.13	96.50 ± 1.46	95.70 ± 1.85
2	2000 × 1000	99.93 ± 0.09	99.49 ± 0.65	99.96 ± 0.05	99.55 ± 0.57	99.49 ± 0.65
5	800 × 600	99.89 ± 0.10	99.19 ± 0.70	99.94 ± 0.05	99.28 ± 0.59	99.18 ± 0.70
10	400 × 300	99.84 ± 0.11	98.85 ± 0.78	99.91 ± 0.06	98.96 ± 0.70	98.84 ± 0.79
20	200 × 150	99.87 ± 0.09	99.08 ± 0.65	99.93 ± 0.05	99.18 ± 0.55	99.08 ± 0.65
30	133 × 100	99.64 ± 0.18	97.47 ± 1.28	99.81 ± 0.10	97.72 ± 1.14	97.45 ± 1.29
38	105 × 78	99.65 ± 0.15	97.52 ± 1.06	99.81 ± 0.08	97.74 ± 0.94	97.50 ± 1.08
40	100 × 75	99.38 ± 0.19	95.64 ± 1.36	99.67 ± 0.10	96.14 ± 1.20	95.62 ± 1.36
50	80 × 60	99.21 ± 0.25	94.39 ± 1.74	99.57 ± 0.13	94.94 ± 1.56	94.31 ± 1.81
60	66 × 50	99.07 ± 0.26	93.43 ± 1.84	99.50 ± 0.14	94.12 ± 1.68	93.32 ± 1.84
70	57 × 42	99.19 ± 0.19	94.23 ± 1.38	99.56 ± 0.10	94.78 ± 1.27	94.13 ± 1.46
80	50 × 37	98.35 ± 0.30	88.27 ± 2.10	99.11 ± 0.16	89.77 ± 1.90	88.02 ± 2.17
90	44 × 33	97.90 ± 0.36	85.08 ± 2.58	98.87 ± 0.20	87.07 ± 2.09	84.97 ± 2.62
100	40 × 30	97.45 ± 0.35	81.95 ± 2.48	98.63 ± 0.19	85.00 ± 2.11	81.73 ± 2.61

Tabela 18 – Desempenho do classificador SVM por classe, utilizando o conjunto de atributos C3 e imagens redimensionadas por $M = 38$ (105×78) *pixels*.

Classe	Métricas				
	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score
1	99.95 ± 00.22	99.27 ± 03.10	100.00 ± 00.00	100.00 ± 00.00	99.61 ± 01.68
2	99.95 ± 00.18	99.27 ± 02.49	100.00 ± 00.00	100.00 ± 00.00	99.62 ± 01.30
3	99.99 ± 00.09	99.82 ± 01.29	100.00 ± 00.00	100.00 ± 00.00	99.90 ± 00.67
4	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00
5	99.00 ± 00.64	96.73 ± 04.41	99.17 ± 00.64	90.43 ± 07.04	93.27 ± 04.14
6	99.99 ± 00.09	100.00 ± 00.00	99.99 ± 00.10	99.83 ± 01.18	99.91 ± 00.61
7	99.55 ± 00.43	96.18 ± 05.83	99.81 ± 00.37	97.72 ± 04.30	96.76 ± 03.19
8	99.96 ± 00.15	100.00 ± 00.00	99.96 ± 00.17	99.50 ± 02.00	99.74 ± 01.04
9	99.64 ± 00.41	98.18 ± 03.67	99.75 ± 00.36	96.98 ± 04.36	97.49 ± 02.88
10	99.95 ± 00.18	100.00 ± 00.00	99.94 ± 00.19	99.33 ± 02.28	99.65 ± 01.19
11	99.68 ± 00.47	96.73 ± 06.56	99.90 ± 00.24	98.83 ± 02.92	97.61 ± 03.70
12	99.77 ± 00.40	100.00 ± 00.00	99.75 ± 00.44	97.10 ± 04.99	98.46 ± 02.67
13	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00
14	98.99 ± 00.70	88.00 ± 09.62	99.82 ± 00.39	97.74 ± 04.71	92.23 ± 05.73
15	98.33 ± 00.93	88.55 ± 08.77	99.08 ± 00.78	88.69 ± 08.64	88.23 ± 06.61
Média	99.65 ± 00.15	97.52 ± 01.06	99.81 ± 00.08	97.74 ± 00.94	97.50 ± 01.08

5.1.2.2 LBP

Do mesmo modo, foi investigado a influência da resolução das imagens no desempenho de acurácia do classificador SVM combinado com o LBP. Os experimentos foram realizados em ambas aplicações.

Resultados na aplicação AP1 (Classificação de Couro)

Na aplicação AP1, também foi avaliado o desempenho do classificador SVM combinado com o extrator LBP usando imagens redimensionadas por M . Neste caso, para os diferentes tamanhos das imagens foram extraídos os atributos de B1, B2, B3 e B4.

A Figura 20 apresenta a variação no desempenho do classificador SVM conforme varia o valor de M . Para este estudo foram utilizados os conjuntos de atributos B1, B2, B3 e B4.

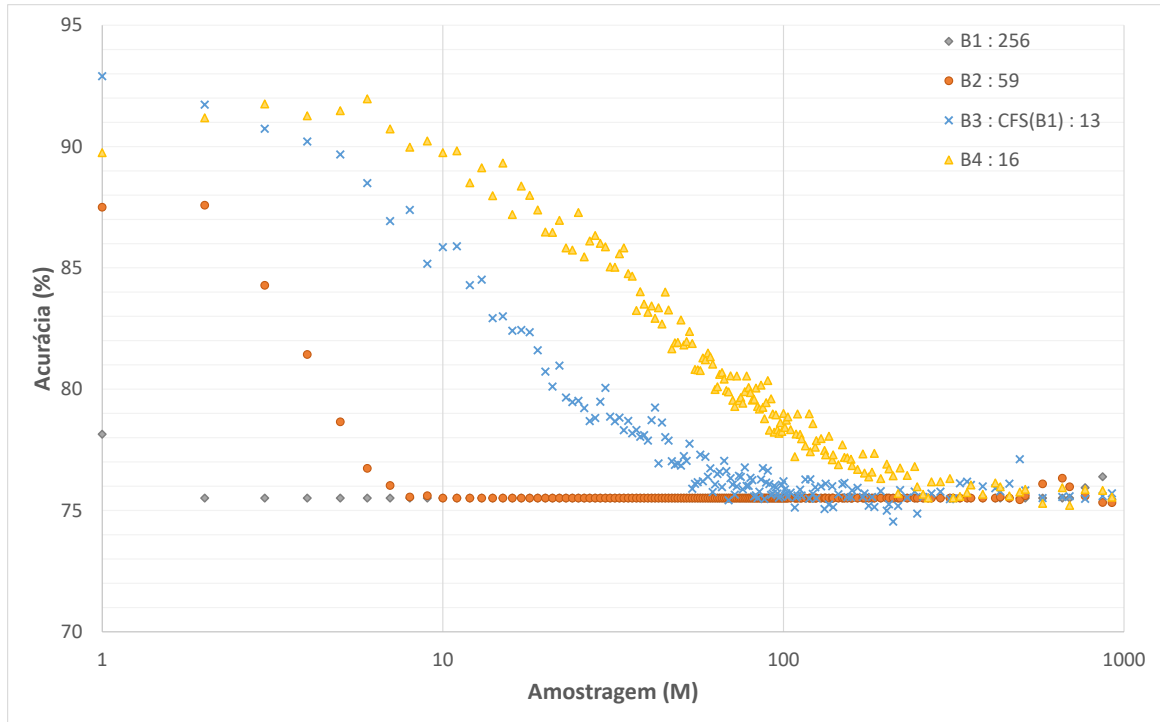


Figura 20 – Desempenho do classificador com LBP para a aplicação AP1 utilizando imagens redimensionadas.

Neste experimento foi verificado que o uso dos conjuntos B1 (256 atributos) e B2 (59 atributos) foram bastantes sensíveis à variação do tamanho da imagem. Com o uso dos 13 atributos do conjunto B3 (CFS em B1), o desempenho do classificador foi melhorado, permitindo o uso de imagens com $M \leq 10$ (345×460) obtendo acurácias entre 85% e 92%. O melhor desempenho foi com o conjunto B4, que obteve acurácias acima de 90% mesmo usando apenas 16 atributos. Aparentemente, com o LBP, quanto menor o número de atributos, menos a sensibilidade do classificador à redução na resolução da imagem.

A Tabela 19 mostra os valores médios e os desvios padrões da métricas obtidas pela combinação do classificador SVM com o extrator LBP utilizando os atributos de B4 para imagens redimensionadas da aplicação AP1.

Na Tabela 20 está apresentado o desempenho do classificador para cada classe. Para esses resultados utilizou-se os atributos de B4 e imagens redimensionadas com $M = 10$ (345×460). Com esta configuração o classificador obteve um desempenho de reconhecimento para os padrões da classe 7 inferior às demais, sendo de aproximadamente 81%.

Tabela 19 – Desempenho do classificador SVM combinado com o extrator LBP usando o conjunto de atributos B4 na aplicação AP1.

M	Tamanho	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score(%)
1	3456×4608	$89,75 \pm 0,98$	$64,11 \pm 3,44$	$94,02 \pm 0,57$	$68,76 \pm 4,81$	$62,11 \pm 3,84$
2	1728×2304	$91,18 \pm 1,18$	$69,14 \pm 4,13$	$94,86 \pm 0,69$	$71,53 \pm 4,44$	$68,44 \pm 4,19$
5	691×921	$91,48 \pm 1,10$	$70,18 \pm 3,86$	$95,03 \pm 0,64$	$71,15 \pm 4,10$	$69,73 \pm 3,79$
10	345×460	$89,75 \pm 1,63$	$64,11 \pm 5,70$	$94,02 \pm 0,95$	$67,90 \pm 5,08$	$64,71 \pm 5,58$
20	172×230	$86,47 \pm 1,06$	$52,66 \pm 3,70$	$92,11 \pm 0,62$	$60,16 \pm 3,66$	$54,23 \pm 3,47$
30	115×153	$85,87 \pm 1,34$	$50,55 \pm 4,68$	$91,76 \pm 0,78$	$61,98 \pm 4,84$	$52,45 \pm 4,59$
40	86×115	$83,17 \pm 1,09$	$41,10 \pm 3,82$	$90,18 \pm 0,64$	$54,79 \pm 5,73$	$42,85 \pm 3,88$
50	69×92	$82,85 \pm 1,20$	$39,98 \pm 4,19$	$90,00 \pm 0,70$	$58,12 \pm 5,93$	$41,59 \pm 4,55$
60	57×76	$81,49 \pm 1,34$	$35,23 \pm 4,69$	$89,21 \pm 0,78$	$53,41 \pm 7,78$	$36,16 \pm 5,21$
70	49×65	$80,55 \pm 1,22$	$31,93 \pm 4,27$	$88,66 \pm 0,71$	$51,85 \pm 7,20$	$32,39 \pm 4,73$
80	43×57	$79,86 \pm 1,22$	$29,52 \pm 4,28$	$88,25 \pm 0,71$	$48,96 \pm 7,31$	$29,74 \pm 4,90$
90	38×51	$80,36 \pm 1,12$	$31,25 \pm 3,92$	$88,54 \pm 0,65$	$48,41 \pm 8,12$	$30,64 \pm 4,35$
100	34×46	$79,00 \pm 1,01$	$26,51 \pm 3,52$	$87,75 \pm 0,59$	$43,31 \pm 10,46$	$25,01 \pm 4,28$

Tabela 20 – Desempenho do classificador SVM por classe, utilizando o conjunto de atributos B4 e imagens redimensionadas por $M = 10$ (345×460) pixels.

Classe	Métricas				
	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score
1	$87,52 \pm 03,37$	$59,69 \pm 11,48$	$92,15 \pm 03,33$	$57,24 \pm 11,82$	$57,82 \pm 10,13$
2	$90,92 \pm 02,29$	$66,92 \pm 12,06$	$94,92 \pm 02,37$	$69,90 \pm 09,67$	$67,56 \pm 08,38$
3	$90,18 \pm 02,93$	$58,77 \pm 14,97$	$95,41 \pm 02,49$	$69,01 \pm 12,49$	$62,54 \pm 12,13$
4	$88,79 \pm 02,20$	$52,62 \pm 13,58$	$94,82 \pm 01,74$	$62,93 \pm 09,31$	$56,60 \pm 10,72$
5	$93,58 \pm 01,97$	$71,69 \pm 11,04$	$97,23 \pm 01,71$	$82,03 \pm 09,60$	$75,91 \pm 08,13$
6	$95,67 \pm 01,99$	$76,46 \pm 12,18$	$98,87 \pm 01,29$	$92,56 \pm 08,15$	$83,07 \pm 08,35$
7	$81,56 \pm 04,61$	$62,62 \pm 12,72$	$84,72 \pm 04,95$	$41,58 \pm 09,25$	$49,48 \pm 09,40$
Média	$89,75 \pm 01,63$	$64,11 \pm 05,70$	$94,02 \pm 00,95$	$67,90 \pm 05,08$	$64,71 \pm 05,58$

Resultados na aplicação AP2 (Localização do Robô)

Do mesmo modo, na aplicação AP2, foi avaliado o desempenho do classificador SVM combinado com o extrator LBP usando imagens redimensionadas por M . Neste experimento foram utilizados os atributos LBP de D1, D2, D3 e D4.

A Figura 21 apresenta a variação no desempenho do classificador SVM com LBP na aplicação AP2 para diferentes valores de M . Neste caso foram utilizados os conjuntos de atributos D1, D2, D3 e D4.

O desempenho do classificador com os 256 atributos de D1, também foi bastante sensível à variação do tamanho da imagem. Com o uso dos padrões binários uniformes presentes em D2 (59 atributos), o classificador obteve bons desempenhos, alcançando valores de $M \leq 30$ com acurácia acima de 95%. O desempenho com D4 (16 atributos), possibilita o uso de valores de $M \leq 100$ (40×30), obtendo acurácias superiores a 97%.

Os valores médios e os desvios padrões da métricas obtidas pela combinação do classificador SVM com o extrator LBP estão apresentados na Tabela 21. Neste caso utilizam-se os atributos de D4 para imagens redimensionadas da aplicação AP2.

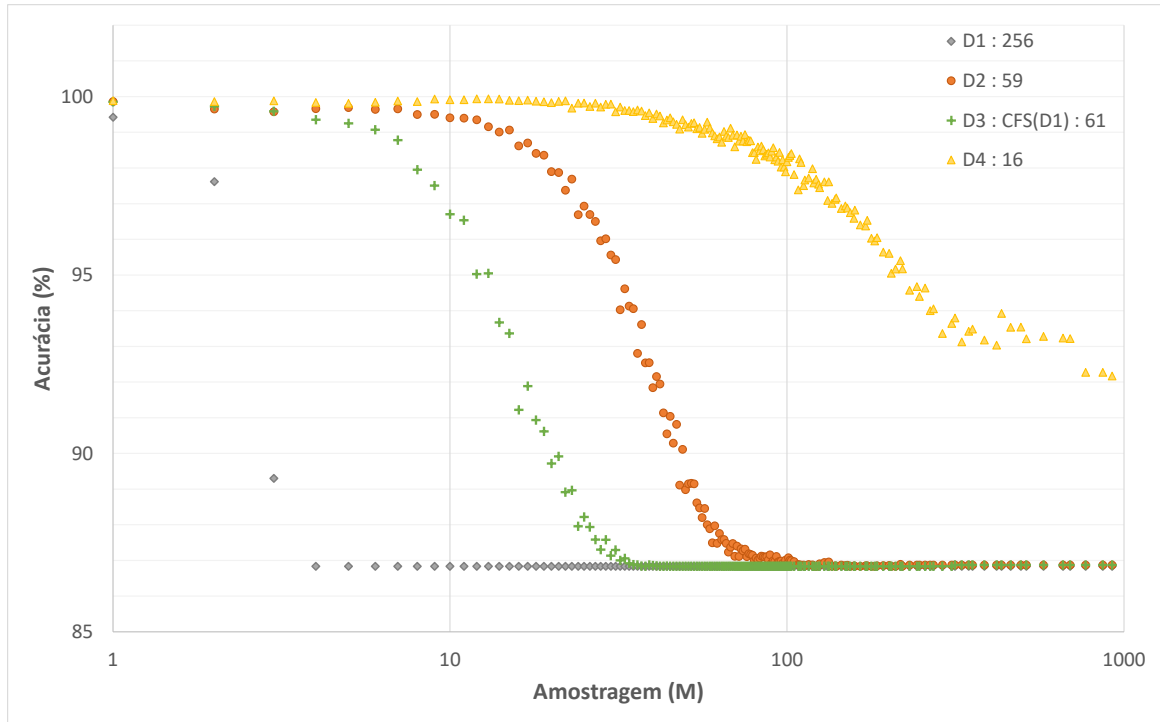


Figura 21 – Desempenho do classificador com LBP para a aplicação AP2 utilizando imagens redimensionadas.

Tabela 21 – Desempenho do classificador SVM combinado com o extrator LBP usando o conjunto de atributos D4 na aplicação AP2.

M	Tamanho	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score(%)
1	4000 × 3000	99,89 ± 0,10	99,20 ± 0,71	99,94 ± 0,05	99,29 ± 0,61	99,20 ± 0,72
2	2000 × 1500	99,86 ± 0,10	99,03 ± 0,69	99,93 ± 0,05	99,13 ± 0,61	99,03 ± 0,69
5	800 × 600	99,81 ± 0,11	98,65 ± 0,79	99,90 ± 0,06	98,93 ± 0,54	98,69 ± 0,74
10	400 × 300	99,91 ± 0,08	99,37 ± 0,56	99,95 ± 0,04	99,45 ± 0,47	99,38 ± 0,55
20	200 × 150	99,83 ± 0,12	98,80 ± 0,84	99,91 ± 0,06	98,92 ± 0,74	98,80 ± 0,84
30	133 × 100	99,79 ± 0,13	98,52 ± 0,93	99,89 ± 0,07	98,82 ± 0,65	98,56 ± 0,89
40	100 × 75	99,39 ± 0,19	95,66 ± 1,37	99,67 ± 0,10	96,73 ± 0,91	95,84 ± 1,27
50	80 × 60	99,21 ± 0,25	94,41 ± 1,80	99,58 ± 0,14	95,80 ± 1,32	94,63 ± 1,70
60	66 × 50	98,98 ± 0,29	92,78 ± 2,09	99,45 ± 0,16	95,03 ± 1,27	93,19 ± 1,87
70	57 × 42	98,59 ± 0,30	90,01 ± 2,09	99,24 ± 0,16	93,01 ± 1,49	90,58 ± 1,91
80	50 × 37	98,43 ± 0,32	88,87 ± 2,25	99,16 ± 0,17	93,57 ± 1,29	89,99 ± 1,93
90	44 × 33	98,47 ± 0,29	89,16 ± 2,03	99,18 ± 0,15	92,94 ± 1,21	89,87 ± 1,78
100	40 × 30	98,17 ± 0,29	87,05 ± 2,04	99,02 ± 0,15	92,34 ± 1,38	88,25 ± 1,76

O desempenho do classificador para cada uma das 15 classes é demonstrado na Tabela 22. Para esses resultados utilizou-se os atributos de D4 e imagens redimensionadas com $M = 10$ (400×300). Sendo assim, o classificador obteve 100% de acurácia nos reconhecimentos de padrões em parte das classes.

Tabela 22 – Desempenho do classificador SVM por classe, utilizando o conjunto de atributo D4 e imagens redimensionadas por $M = 10$ (40×30) *pixels*.

Classe	Métricas				
	Ac(%)	Se(%)	Es(%)	VPP(%)	F1-Score
1	99,91 ± 00,22	98,73 ± 03,19	100,00 ± 00,00	100,00 ± 00,00	99,33 ± 01,67
2	99,92 ± 00,21	100,00 ± 00,00	99,92 ± 00,23	99,00 ± 02,74	99,48 ± 01,43
3	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
4	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
5	99,81 ± 00,30	97,27 ± 04,21	100,00 ± 00,00	100,00 ± 00,00	98,57 ± 02,20
6	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
7	99,81 ± 00,30	97,27 ± 04,21	100,00 ± 00,00	100,00 ± 00,00	98,57 ± 02,20
8	99,86 ± 00,27	98,00 ± 03,80	100,00 ± 00,00	100,00 ± 00,00	98,95 ± 01,99
9	99,99 ± 00,09	99,82 ± 01,29	100,00 ± 00,00	100,00 ± 00,00	99,90 ± 00,67
10	99,96 ± 00,15	99,45 ± 02,18	100,00 ± 00,00	100,00 ± 00,00	99,71 ± 01,14
11	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
12	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
13	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
14	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00	100,00 ± 00,00
15	99,41 ± 00,55	100,00 ± 00,00	99,37 ± 00,59	92,72 ± 06,32	96,11 ± 03,47
Média	99,91 ± 00,08	99,37 ± 00,56	99,95 ± 00,04	99,45 ± 00,47	99,38 ± 00,55

5.1.2.3 Discussão parcial dos resultados

Nos experimentos realizados com as imagens reduzidas, a utilização do extrator GLCM na aplicação AP2 obteve desempenhos de acurácia superiores comparados com a AP1. Isso pode ser justificado pelo alto grau de superposição entre as classes na aplicação AP1. Com esse mesmo extrator, o uso de imagens com tamanhos menores para $M > 100$, leva o classificador a cometer erros elevados em ambas as aplicações.

O uso dos 16 atributos LBP de B4 e D4, obtiveram bons desempenhos de acurácia para valores de $M \leq 10$. Na aplicação AP1 com imagens reduzidas até 345×460 *pixels*, foram obtidas acurácias em torno de 90%. Por outro lado, na AP2 obteve-se acurácias próximas a 100% com imagens reduzidas até 400×300 *pixels*.

No geral, a aplicação AP1 apresentou maior sensibilidade à redução da resolução das imagens do que a aplicação AP2. Isso pode ser justificado pelas características inerentes de cada base e dados.

Com esses experimentos, é possível verificar o impacto que o tamanho das imagens pode causar no sistema e escolher um conjunto de atributos adequado à aplicação. Esse estudo também vai permitir a escolha do tamanho de imagem ideal para ser usado na plataforma embarcada.

5.1.3 Escolha do tamanho ideal da imagem e do conjunto de atributos

Esta seção visa a um processo para escolher o valor de M ideal no redimensionamento das imagens foi realizado. Este procedimento se baseia no desempenho do classificador nas aplicações. O valor de M ideal é o fator de redução das imagens, no qual o classificador SVM obtém, em comum, bons desempenhos de acurácia na utilização dos

conjuntos de atributos para cada extrator.

Partindo do entendimento que o desempenho do classificador é melhor usando as imagens no seu tamanho original do que utilizando imagens redimensionadas. Obtém-se uma média entre os valores de taxas de acerto para $1 \leq M \leq 5$ utilizando cada conjunto de atributos para 5 rodadas. Em cada rodada o classificador foi treinado e testado 50 vezes com $1 \leq M \leq 1000$. Cria-se uma lista para cada conjunto de atributos contendo os valores de M que obtiveram taxas de acerto maior ou igual à média. Ao fim é feita uma intersecção destas listas e o maior valor é o melhor M .

A Tabela 23 apresenta os conjuntos de atributos e tamanhos das imagens considerados ideais para cada extrator nas aplicação AP1 e AP2. Para o LBP foram considerados somente os conjuntos de atributos que não apresentaram desempenhos sensíveis ao tamanho da imagem. Sendo assim, os desempenhos com todos os 256 atributos do LBP original e os 59 dos padrões binários uniformes não foram postos para a escolha do valor de M ideal.

Para a escolha dos conjuntos de atributos considerados ideais em cada aplicação, são eleitos os conjuntos que apresentam os seguintes resultados: menor quantidade de atributos (Seção 4.4), menor tempo de computação na extração dos atributos (Seção 5.1.1), e bons desempenhos de acurácia para imagens reduzidas (Seção 5.1.2).

Tabela 23 – Conjuntos de atributos e tamanhos das imagens considerados ideais para as aplicações.

Aplicação	Extrator	Atributos	Tamanho ideal	Ac(%)
AP1 (Classificação de Couro)	GLCM	A3	M = 73 (47 × 63)	90,33 ± 1,12
	LBP	B4	M=10 (345 × 460)	89,75 ± 1,63
AP2 (Localização do Robô)	GLCM	C3	M=38 (105 × 78)	99,65 ± 0,15
	LBP	D4	M=10 (400 × 300)	99,91 ± 0,08

5.1.4 Consumo de memória

Nesta seção foi elaborado um estudo analítico para estimar o consumo de memória das técnicas utilizadas nas aplicações. Com esse estudo é possível antecipar o impacto em relação ao gasto de memória em uma plataforma embarcada.

5.1.4.1 Estudo analítico

O estudo foi aplicado com o intuito de obter um Valor Estimativo de Consumo de Memória (VECM) das técnicas GLCM, LBP e SVM. As medidas do VECM em *bits* são definidas conforme as equações abaixo.

Notação:

- N_g : Número de níveis de cinza da GLCM;
- b : Número de *bits* referente ao tipo de dado utilizado;
- w : Quantidade de atributos;
- P : Número de *pixels* vizinhos no LBP;
- L : Quantidade de linhas na imagem;
- C : Quantidade de colunas na imagem;
- $\log(L.C)$: Número de *bits* baseado na resolução da imagem;
- n : Quantidade de classes do problema;
- V : $\sum_{i=1}^n v_i$, somatório da quantidade de vetores de suportes v_i das n classes;

Equações:

$$VECM_{(GLCM)} = (N_g^2 + w).b \quad (5.1)$$

$$VECM_{(LBP)} = P.b + w.\log(L.C) \quad (5.2)$$

$$VECM_{(SVM)} = (V.w + V + n)b \quad (5.3)$$

Baseada nas equações acima, a Tabela 24 apresenta os consumos estimativos de memória, em *kilobytes* (kB), das técnicas GLCM, LBP e SVM na aplicação AP1 utilizando imagens reduzidas. Para o extrator GLCM foram usados os 10 atributos de A3 e para o LBP os 16 atributos de B4. Também foram mensurados os consumos de memória da SVM usando os atributos A3 (GLCM) e B4 (LBP).

Tabela 24 – Consumos estimativos de memória (em *kilobytes*) das técnicas GLCM, LBP e SVM na aplicação AP1 utilizando imagens redimensionadas. V representa o total de vetores de suporte.

Técnica	M=1 (3456 × 4608)	M=10 (345 × 460)	M=50 (69 × 92)	M=73 (47 × 63)	M=100 (34 × 46)
GLCM	256,039				
LBP	0,018	0,014	0,010	0,010	0,010
SVM (A3:GLCM)	16,367 $V = 419$	15,469 $V = 396$	17,617 $V = 451$	17,773 $V = 455$	19,648 $V = 503$
SVM (B4:LBP)	29,938 $V = 479$	35,250 $V = 564$	70,625 $V = 1130$	78,625 $V = 1258$	80,063 $V = 1281$

Do mesmo modo, a Tabela 25 apresenta os consumos estimativos de memória, em *kilobytes* (kB), das técnicas GLCM, LBP e SVM na aplicação AP2 utilizando imagens reduzidas. Para o extrator GLCM foram usados os atributos de C3 e para o LBP os atributo de D4. Para a SVM foram mensurados os consumos de memória utilizando os atributos C3 (GLCM) e D4 (LBP).

O consumo de memória utilizando a GLCM, independentemente do tamanho da imagem, está diretamente relacionado com o tamanho da MC e com a quantidade

Tabela 25 – Consumos estimativos de memória (em *kilobytes*) das técnicas GLCM, LBP e SVM na aplicação AP2 utilizando imagens redimensionadas. V representa o total de vetores de suporte.

Técnica	M=1 (4000 × 3000)	M=10 (400 × 300)	M=38 (105 × 78)	M=50 (80 × 60)	M=100 (40 × 30)
GLCM	256,039				
LBP	0,018	0,014	0,010	0,010	0,010
SVM (C3:GLCM)	12,734 $V = 326$	11,445 $V = 293$	11,563 $V = 296$	13,281 $V = 340$	18,672 $V = 478$
SVM (D4:LBP)	19,188 $V = 307$	34,625 $V = 554$	59,250 $V = 948$	62,313 $V = 997$	79,813 $V = 1277$

de atributos selecionados. O tamanho da MC se refere à quantidade de níveis de cinza considerados nas imagens. Para uma MC considerando todos os 256 níveis de cinza e com a utilização dos 10 atributos de A3 (ou C3), foi obtido um consumo de aproximadamente 256,039 kB. Isso considerando o tipo de dado de todos elementos da MC normalizada e todos os atributos como ponto flutuante ($b = 32 \text{ bits}$).

Os consumos de memória do extrator LBP foram semelhantes para ambas aplicações. Com a utilização dos 16 atributos do conjunto B4 (ou D4) representados por inteiros de $b = 8 \text{ bits}$, os consumos foram abaixo de 1 kB.

Em relação ao consumo de memória do classificador SVM após ao seu processo de treinamento com os extratores GLCM e LBP, o principal fator de consumo é a quantidade de vetores de suportes V necessários para o reconhecimento de padrões. Baseado nas Tabelas 24 e 25, mesmo com um problema de 15 classes, os consumos dos modelo do classificador SVM na aplicação AP2 obtiveram valores menores comparados com a aplicação AP1 de 7 classes. Além dos tamanhos das imagens da aplicação AP2 serem menores do que as imagens da AP1, isso também pode ser devido à base de dados da aplicação AP1 apresentar alto grau de superposição entre as classes, acarretando pequenas variações inter-classes. Por consequência o modelo precisará de uma representação maior de vetores de suporte na tentativa de classificar (separar) os padrões.

Vale ressaltar que os valores dos consumos de memória do classificador SVM foram extraídos de modelos gerados na ET1 com a ferramenta Matlab. Foram escolhidos os modelos do classificador SVM que obtiveram melhores desempenhos de acurácia nas 50 rodadas para diferentes tamanhos de imagens. O método abordado na classificação multiclasse foi o Um-Contra-Todos (UCT) e todos os elementos dos vetores de suportes são do tipo ponto flutuante ($b = 32 \text{ bits}$).

Em suma, a partir dos resultados, constatou-se que o uso dos atributos LBP (B4 e D4 de 16 atributos) consome menos memória que com os atributos GLCM (A3 e C3 de 10 atributos). Foi verificado que MC é um fator impactante no consumo de memória na extração de atributos com a GLCM. Entretanto, a redução da MC pode ser uma solução viável na diminuição do consumo de memória.

5.1.4.2 Redução da Matriz de Coocorrência (MC)

Nos experimentos até aqui a MC utilizou $N_g = 256$ níveis de cinza encontrados nas imagens com resolução de 8 *bits*. De acordo com o estudo do consumo estimativo de memória apresentado anteriormente. Para extrair os 10 atributos de A3 ou C3 com a GLCM, são necessários 256,039 kB de memória RAM. A maior parte desse consumo se deve ao tamanho da MC (256×256).

Com o objetivo de diminuir o espaço de armazenamento ocupado na memória pela CM no ambiente embarcado, uma avaliação de desempenho do classificador foi realizada para diferentes valores de N_g . Os gráficos de dispersão da Figura 22 apresenta as taxas de acerto do classificador para cada aplicação com a seguinte configuração: imagens redimensionadas com $1 \leq M \leq 100$, conjunto de atributos A3 e redução da MC por $F = \{1, 2, 4, 8, 16\}$. Para cada combinação dos valores de F e M foi obtido um valor da taxa de acerto, resultando em 100 valores para cada um dos 5 grupos de dispersão.

Nestes resultados, para os valores de $F = \{1, 2, 4, 8, 16\}$, tem-se na ordem os valores $N_g = \{256, 128, 64, 32, 16\}$. Em resoluções efetivas, as imagens foram reduzidas de 3456×4608 ($M = 1$) a 34×46 ($M = 100$) para a aplicação AP1 e 4000×3000 ($M = 1$) a 40×30 ($M = 100$) para a AP2.

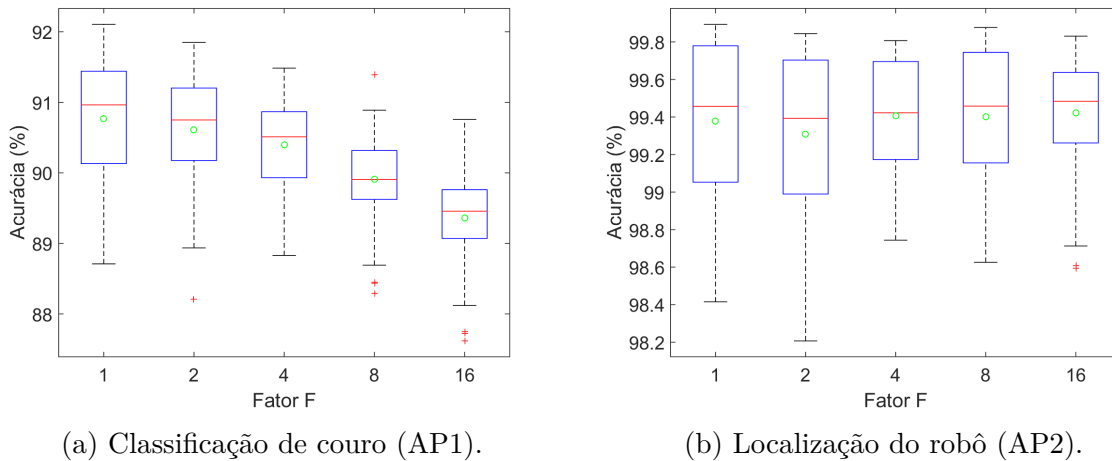


Figura 22 – Desempenho do classificador com a GLCM redimensionada.

No experimento com as imagens de couro (Figura 22a) a redução da matriz de coocorrência provocou uma queda no desempenho do classificador. Já com as imagens da localização do robô móvel (Figura 22b) o desempenho apresentou um comportamento constante.

A diferença nos resultados para as aplicações se justifica pela seguinte análise. Verificando o histograma de uma amostra para cada base de dados das aplicações, as imagens de couro (Figura 23a), considerando somente a ROI, apresentam quantidades de níveis de cinza concentrados praticamente em uma região enquanto no histograma

da outra base (Figura 23b), espalhados. Sendo assim, nas imagens da aplicação AP2, na medida que for diminuindo a quantidade de níveis de cinza considerados na imagem para a MC, ainda se tem informação necessária para o bom desempenho do classificador diferente do que ocorre com as imagens da aplicação AP1.

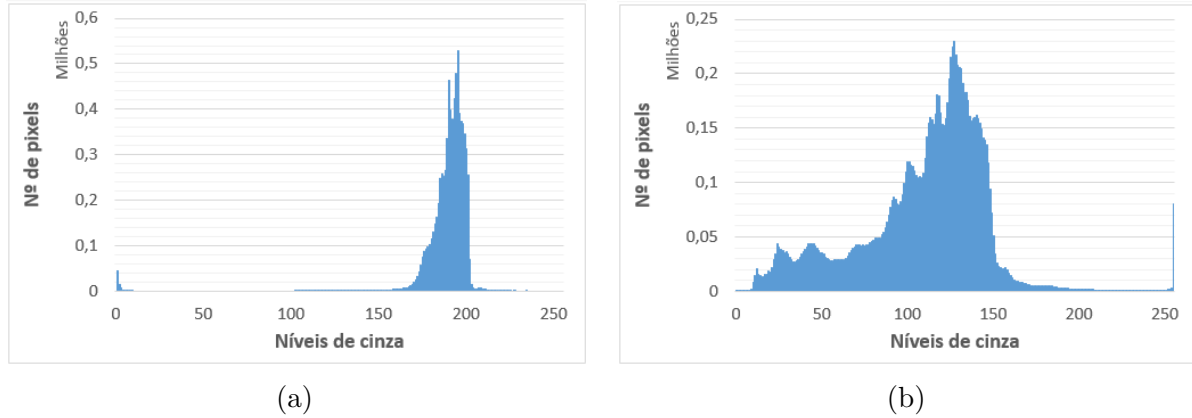


Figura 23 – Histograma de imagens das aplicações AP1 e AP2, respectivamente.

Contudo, o ganho de memória e tempo de computação dos atributos obtidos pela redução da MC é vantajoso. Sendo assim, na aplicação AP1 permite-se utilizar a MC com dimensões de 64×64 ($N_g = 64$), obtendo acurácias em torno de 90% e consumo de 16,039 kB de memória. Já na AP2 pode-se utilizar a MC de tamanho 16×16 ($N_g = 16$), mantendo acurácias acima de 99% e consumo de memória de apenas 1,039 kB. Perante isso, a redução da MC é uma abordagem interessante na tentativa de reduzir o consumo de memória obtido com a utilização do extrator GLCM.

5.2 Resultados na plataforma embarcada

Foram embarcadas as etapas de extração de atributos presentes em A3, B1, C3 e D4, e o processo de classificação. Sendo assim, o sistema é testado com uma imagem contendo somente a ROI, processada na estação de trabalho ET1 e armazenada no cartão de memória da plataforma.

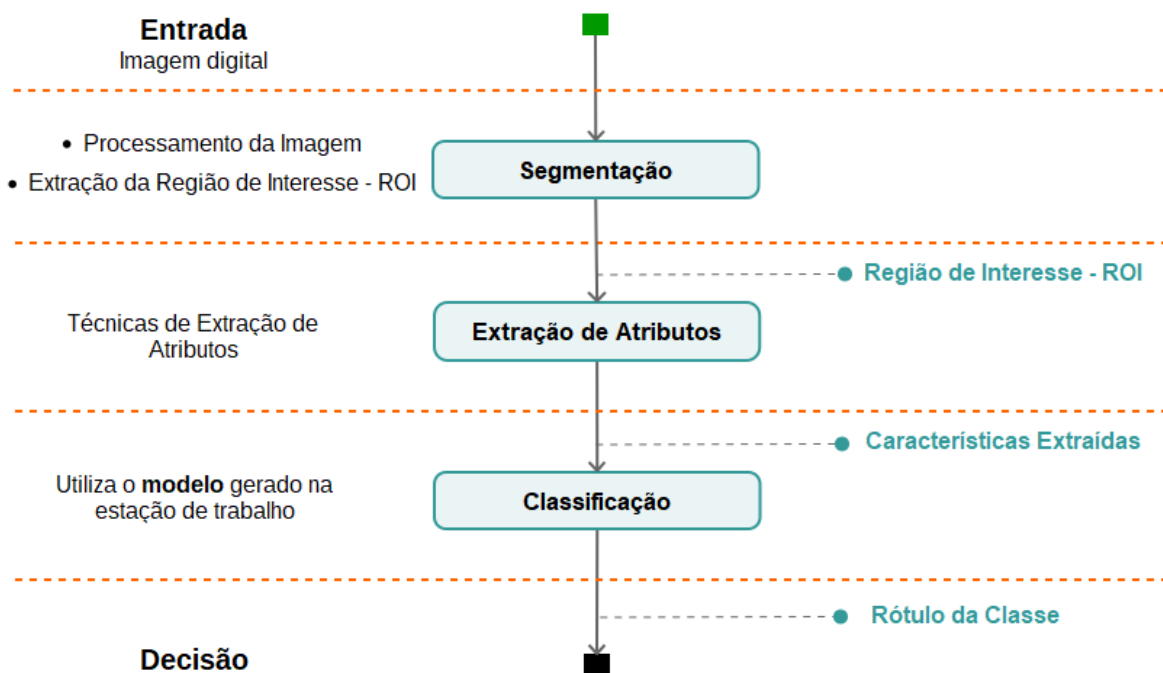
A avaliação das restrições de embarcar o processo de classificação, será baseado no sistema de reconhecimento de padrões proposto na Figura 24. O objetivo desta avaliação é identificar, no processo de classificação, algumas propriedades que afetam o consumo de recursos na plataforma embarcada. As etapas do sistema de reconhecimento deste trabalho são descritas da seguinte forma:

- **Entrada ou Aquisição:** Refere-se à entrada da imagem no sistema através de uma câmera digital ou armazenada na memória da plataforma embarcada.
- **Segmentação:** Etapa que com a ajuda de técnicas de processamento de imagens, faz a extração da região de interesse (ROI), no caso, a região que define o couro

para a aplicação AP1. Nesta etapa, o couro é separado do restante da imagem de entrada. Na aplicação AP2 a ROI é toda a imagem de entrada.

- **Extração de Atributos:** A partir da ROI é possível extrair as características que podem revelar um padrão na imagem. Tem como objetivo caracterizar um objeto para ser reconhecido por medidas que são muito parecidas em objetos da mesma classe e muito diferentes para objetos de outras classes.
- **Classificação:** A partir de um modelo definido no processo de treinamento em ambiente computacional, o classificador utiliza as características extraídas para determinar a qual classe pertence o padrão.
- **Decisão:** Executa a ação adequada (por exemplo, acender LEDs) em função do rótulo da classe resultante do processo de classificação.

Figura 24 – Sistema de Reconhecimento de Padrões usado no trabalho em uma plataforma embarcada.



O sistema de reconhecimento de padrões foi implantado na plataforma embarcada Raspberry Pi 3 *Model B*, que contém uma CPU de 1.2GHz 64-bit quad-core ARMv8, memória RAM de 1GB e fornece entrada para cartão de memória não volátil micro SD, onde o sistema operacional é instalado. O sistema operacional utilizado foi o Raspbian, um sistema operacional livre baseado em Linux Debian. A codificação do sistema foi implementada utilizando a biblioteca OpenCV com suporte as linguagens de programação C, C++ e Python. Esta placa foi escolhida como primeiro ambiente de validação pela sua disponibilidade a um baixo custo.

O classificador SVM foi treinado em ambiente computacional na estação de trabalho ET2 (Linux + OpenCV), apresentando resultados de desempenho iguais à estação ET1 (Windows + Matlab). O *kernel* usado no SVM foi o RBF com um parâmetro manual de $\sigma = 2$. No treinamento foram utilizadas 50 realizações (rodadas). O modelo com os melhores resultados de acurácia foi exportado para a Raspberry.

5.2.1 Validação de resultados

Para comparar os resultados do classificador SVM em uma estação de trabalho com os resultados da plataforma embarcada, foi feito o seguinte experimento. Todos os 50 conjuntos de dados para testes, incluindo os rótulos de saída do modelo e os vetores de suporte do processo de classificação utilizados na ET2, foram armazenados em arquivos e posteriormente carregados e executados na plataforma embarcada. Para cada entrada processada pela plataforma embarcada foi verificado se os rótulos de saída correspondiam com os rótulos extraídos na ET2. Neste experimento, foi constatado que os resultados de desempenho do classificador SVM na ET2 e na plataforma embarcada foram iguais.

Os vetores de suporte junto com as funções de decisão e os valores de *alpha* e *bias*, os quais representam o modelo definido no processo de aprendizagem (treinamento) do classificador, foram implantados no ambiente embarcado. A implantação foi realizada com recursos da própria OpenCV, que permite exportar esses dados na máquina de treinamento. Foram embarcados somente os modelos do classificador SVM que obtiveram melhores desempenhos de acurácia nas 50 rodadas.

5.2.2 Medições de tempo de computação

Para medir o tempo de computação na plataforma embarcada, para cada aplicação e extrator de atributo, um experimento foi realizado utilizando uma amostra de imagem da classe 1 armazenada no cartão de memória.

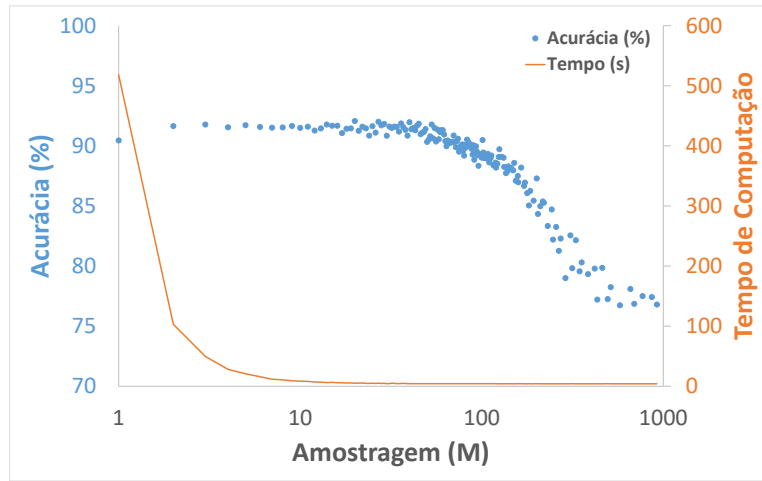
No total foram executadas 25 medições em cada tamanho de imagem para determinar o tempo de computação na plataforma embarcada. A cada 5 medidas foi calculada uma média e das 5 médias foi escolhido o menor valor.

Os consumos de tempos de computação foram avaliados com as imagens redimensionadas. Na medição usando a GLCM foram consideradas as seguintes etapas: carregamento da imagem, criação e normalização da matriz MC, extração de características e classificação. Já com o LBP foram avaliados o carregamento da imagem, extração de características e classificação. As medidas foram realizadas para demonstrar o ganho de tempo de computação e economia de memória que uma imagem redimensionada pode ocasionar.

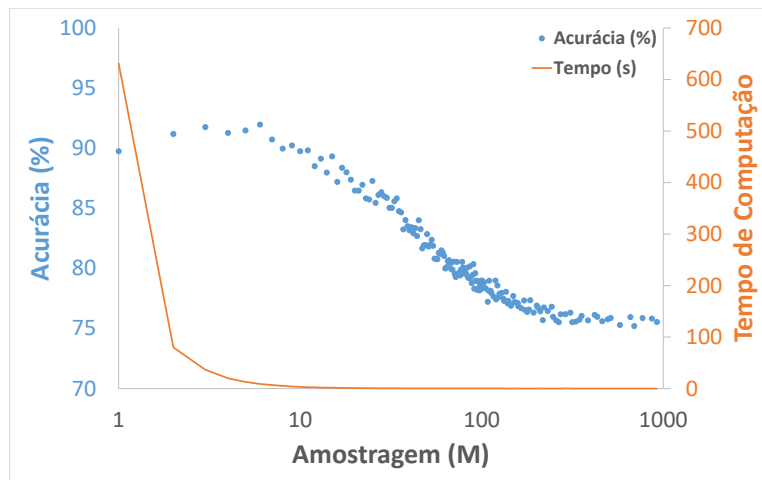
Posto isto, os resultados de tempo de computação e consumo de memória para cada aplicação são apresentados logo a seguir.

5.2.3 Resultados na aplicação AP1

A Figura 25 apresenta os tempos totais de computação obtidos na plataforma embarcada para todas as imagens redimensionadas na classificação de couro caprino. Tanto para o uso do conjunto A3 (Figura 25a) quanto para B4 (Figura 25b), os gráficos mostram uma forte redução no tempo de computação, mesmo com pequenas reduções no tamanho da imagem. Isso indica que se pode diminuir o tempo de processamento sem precisar perder muito em acurácia do classificador. Por outro lado, nota-se que a partir de $M = 10$ (345 x 460) não há redução significativa no tempo, não justificando o uso de imagens menores. A não ser para reduzir o gasto de memória.



(a) A3-GLCM-AP1



(b) B4-LBP-AP1

Figura 25 – Tempos de computação (em segundos) na plataforma embarcada para a aplicação AP1.

A Tabela 26 exhibe, por etapas, o tempo de computação (em segundos) dos processos de extração de atributos e de classificação utilizando os conjuntos de atributos A3 e B4. Com a GLCM, observa-se que a etapa de criar a MC é bastante sensível ao tamanho da imagem, tornando-se desprezível para imagens pequenas. A outra etapa que contribui significativamente no tempo total é o cálculo dos atributos. Essa é dependente do tamanho da matriz de coocorrência e por isso não sofre alteração com o tamanho da imagem. Cabe ainda destacar o baixo valor do tempo de computação da SVM. Isso explica porque todo o esforço pra otimizar o processo recaiu sobre a extração de atributos. Considerando que a indústria aceita um tempo médio de classificação de 31,8s com taxas de acerto maiores que 80% para um perito (MEDEIROS, 2007), a resolução de imagem $47 \times 63 \text{ pixels}$ consumir 3,1s com acurácia de 90% (Tabela 16) é um bom resultado.

Em relação ao LBP, foi obtido um tempo de computação de 6,8s com acurácia de 89% Tabela 20) utilizando imagens com resolução de $345 \times 460 \text{ pixels}$ ($M = 10$). Para imagens menores ($M > 10$), o tempo de computação diminui, entretanto, o desempenho do classificador cai significativamente (Figura 25b).

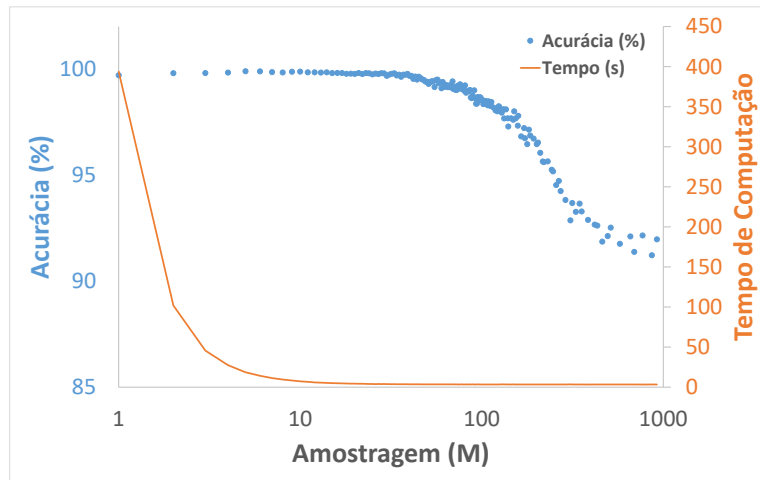
Tabela 26 – Tempo de computação (em segundos) por etapas na aplicação AP1.

AP1 : Classificação de couro caprino						
Extrator	Etapas	M=1 (3456x4608)	M=10 (345x460)	M=50 (69x92)	M=73 (47x63)	M=100 (34x46)
GLCM	Carregar a imagem	0,3020	0,0034	0,0005	0,0002	0,0003
	Criar a matriz MC	514,7374	5,1216	0,2074	0,1000	0,0523
	Normalizar a matriz MC	0,3683	0,3672	0,3505	0,3575	0,3521
	Extrair os atributos	2,6193	2,6173	2,6168	2,6402	2,6161
	Classificar (SVM)	0,0093	0,0091	0,0091	0,0080	0,0091
Total:		518,0363	8,1186	3,1843	3,1069	3,0299
LBP	Carregar a imagem	0,3020	0,0034	0,0005	0,0002	0,0003
	Extrair os atributos	630,6950	6,8464	0,6436	0,5367	0,4708
	Classificar (SVM)	0,0089	0,0089	0,0089	0,0089	0,0089
Total :		631,0059	6,8587	0,6530	0,5458	0,4800

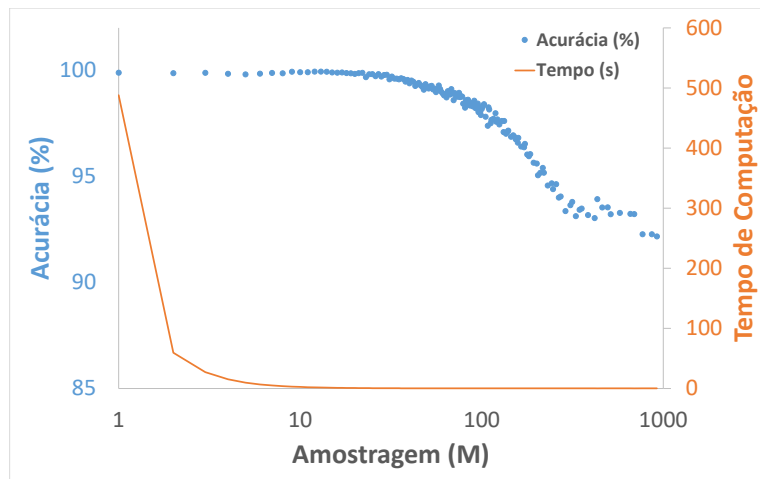
5.2.4 Resultados na aplicação AP2

Do mesmo modo a Figura 26 apresenta os tempos totais de computação obtidos na plataforma embarcada para todas as imagens redimensionadas na aplicação de localização e navegação do robô utilizando os conjuntos C3 (10 atributos) e D4 (16 atributos), com os extratores GLCM e LBP, respectivamente.

O tempo de computação apresentou a mesma característica da AP1, onde a partir de $M = 10$ (400×300) tendeu a ser constante. Com esta análise, na aplicação AP2 pode ser utilizado imagens de tamanhos na faixa de $10 \leq M \leq 100$ (400×300 a 30×40) tendo pouca variação da acurácia do classificador e bons tempos de computação.



(a) C3-GLCM-AP2



(b) D4-LBP-AP2

Figura 26 – Tempos de computação na plataforma embarcada para a aplicação AP2.

O tempo de computação (em segundos) dos processos de extração de atributos e de classificação com os conjuntos C3 (GLCM) e D4 (LBP) estão apresentados na Tabela 27. Nestes resultados foi constatado que com o LBP, o tempo de computação é menor comparado ao GLCM para imagens menores. Justifica-se pelo fato que o tempo da extração de atributos do LBP está mais atrelado ao tamanho da imagem do que na GLCM.

Tabela 27 – Tempo de computação (em segundos) por etapas na aplicação AP2.

AP2 : Localização e navegação do robô móvel						
Extrator	Etapas	M=1 (4000x3000)	M=10 (400x300)	M=38 (105x78)	M=50 (80x60)	M=100 (40x30)
GLCM	Carregar a imagem	0,3590	0,0032	0,0004	0,0003	0,0001
	Criar a matriz MC	382,6410	3,9687	0,2719	0,0155	0,0042
	Normalizar a matriz MC	6,0000	0,4440	0,5544	0,5504	0,5526
	Extrair os atributos	7,0000	2,6260	2,6312	2,6314	2,6412
	Classificar (SVM)	0,0089	0,0089	0,0089	0,0089	0,0089
Total:		396,0089	7,0508	3,4668	3,2065	3,2070
LBP	Carregar a imagem	0,3590	0,0032	0,0004	0,0003	0,0001
	Extrair os atributos	487,6410	5,0567	0,7305	0,5967	0,4478
	Classificar (SVM)	0,0089	0,0090	0,0089	0,0089	0,0089
Total:		488,0089	5,0689	0,7398	0,6059	0,4568

5.3 Análise Geral dos Resultados

Em suma, esse capítulo apresentou os resultados dos testes obtidos pela combinação dos extratores de características GLCM e LBP com o classificador SVM nas tarefas de classificação da qualidade de couro caprino (AP1) e na localização e navegação de um robô móvel (AP2). No estudo foram feitos experimentos para avaliar o tempo de computação, consumo de memória e desempenho do classificador em uma estação de trabalho (ET1). Por fim, os experimentos de tempos de computação com imagens reduzidas são avaliados em uma plataforma embarcada.

Para a classificação das imagens de couro caprino, os tempos de computação obtidos foram de 3,1s e 0,54s, e consumos de memória de 16,039 kB e 0,010 kB com acurácias superiores a 90% na classificação de imagens com resolução 47×63 *pixels* com o uso de GLCM e LBP, respectivamente. Já para a localização do robô móvel foram obtidos tempos de computação de 3,2s e 0,45s, e consumos de memória de 1,039 kB e 0,010 kB com acurácias próximas a 100% utilizando imagens de 40×30 *pixels*. Esses resultados permitem utilizar plataformas baseadas em microcontroladores para aplicações de visão computacional, reduzindo custo de desenvolvimento e de implantação.

Contudo, os resultados obtidos poderiam ser mais precisos com a utilização de uma quantidade maior de amostras nas bases de dados das aplicações estudadas. Visto que para a classificação de couro, foram utilizadas apenas 13 amostras para o teste de classificação do modelo. Já para a localização e navegação do robô foram utilizadas apenas 10 amostras.

No geral, os resultados obtidos na aplicação AP2 foram melhores comparados com os resultados da aplicação AP1. Dentre as razões para os resultados da aplicação AP2, além de apresentar imagens de tamanho menor aos da AP1, especulam-se as seguintes: (i) baixo grau de subjetividade na determinação das classes e (ii) baixo grau de superposição entre as classes. Possibilitando assim, a condução de uma representação compacta das amostras e simplificação da tomada de decisão do classificador.

6 Conclusão

Este trabalho apresenta uma análise das restrições de embarcar uma aplicação de visão computacional, composta de um classificador que combina SVM com os extratores de atributos GLCM e LBP. Esses extratores foram escolhidos por terem sido largamente utilizados em aplicações na literatura, onde a análise e descrição de texturas das imagens são essenciais para um bom desempenho do processo de classificação. São investigadas as relações entre os atributos extraídos da imagem pelos extratores e o consumo de recursos em um computador embarcado, sempre procurando preservar a taxa de acerto do classificador.

Como estudos de caso são usadas duas aplicações, classificação da qualidade de couro caprino e localização e navegação de um robô móvel. Essas duas aplicações foram escolhidas por apresentarem características diferentes tanto no domínio de aplicação quanto na base de dados. A do couro é típica de indústria na detecção de falhas na matéria prima, ou no produto. A outra na tarefa de reconhecimento de locais internos para localização e navegação de um robô móvel usando mapa topológico.

Essa investigação pode contribuir para a criação de um produto final de baixo custo, destinado às indústrias processadoras de couros ou para aplicações de localização e navegação de robôs móveis autônomos.

O classificador SVM, além dos excelentes resultados de acurácias, já consagrados na literatura, tem um tempo de computação baixo. Isso deixa o desafio de otimização nessa área somente para o processo de extração de atributos.

Três explorações para avaliar o consumo de recursos computacionais são apresentadas. A resolução da imagem de entrada, a seleção de atributos e o tamanho da matriz de coocorrência (no caso da GLCM). As métricas para avaliar estas explorações são: tempo de computação e memória, sempre procurando manter a máxima acurácia do classificador.

Na classificação das imagens de couro caprino com a GLCM, na plataforma embarcada o tempo de computação alcançou 3,1 segundos, com taxas de acerto superiores a 90% e imagens com resolução 47×63 *pixel*. A indústria considera que um perito utiliza um tempo médio de classificação de 31,8 s e obtém uma taxa de acerto superior a 80%.

O redimensionamento da imagem não interferiu significativamente nas taxas de acerto do modelo. Ou seja, pode-se escolher um tamanho ideal da imagem para o sistema alvo, operando com baixo tempo de computação e com um menor espaço armazenado na memória.

Foi proposta a seleção de atributos GLCM baseada no estudo de complexidade temporal, constatando-se um resultado de acurácia excelente nas aplicações estudadas.

Outras investigações relacionadas a GLCM não se preocuparam em avaliar os custos computacionais na extração dos atributos e seu impacto no desempenho do classificador.

Foi avaliada uma extensão do LBP que utiliza apenas $P = 4$ *pixels* vizinhos ao invés de 8 *pixels* na sua versão original. Neste caso, além de proporcionar uma menor representatividade de seus atributos na memória e menor tempo de computação, foram obtidos bons valores de acurácia em ambas as aplicações, principalmente para imagens com baixa resolução.

Para o extrator GLCM a redução na matriz de coocorrência é interessante para reduzir o tempo de computação dos atributos. Entretanto, tamanhos menores que 256×256 podem reduzir a acurácia do classificador, se a aplicação fornece imagens com baixa variação de níveis de cinza.

Comparando os resultados com os trabalhos relacionados, na AP1 a taxa de acerto médio do classificador SVM combinado com GLCM para imagens de resolução 34×46 *pixels* foi acima de 89%, superando o desempenho obtido em SANTOS FILHO (2013), que obteve acurácia de aproximadamente 69% (classificação binária) para as mesmas imagens usadas nesta dissertação, mas com resolução superior (100×100 *pixels*). Esta pesquisa alcançou melhores resultados mesmo para um classificador de 7 classes. Essa melhora se deve ao uso de um conjunto de atributos diferente. Em relação ao desempenho de acurácia do SVM combinado com GLCM e LBP na AP2, foram obtidos resultados semelhantes aos apresentados em Marinho (2017), mas com imagens redimensionadas para resoluções menores.

6.0.1 Proposta de trabalhos futuros

O trabalho apresentado nesta dissertação pode ser estendido em várias direções. Como trabalhos futuros, sugerem-se:

- Avaliar a SVM com outros métodos de extração de atributos;
- Aplicar outras técnicas de aprendizagem de máquina;
- Utilizar plataformas embarcadas com menos recursos, visando à redução de custos e/ou aumento da confiabilidade;
- Realizar o mesmo estudo em outras aplicações;
- Desenvolver um processo de aquisição das imagens em tempo real, visando à construção de um protótipo para teste em campo.

Referências

ALBREGTSEN, F. *Statistical Texture Measures Computed from Gray Level Cooccurrence Matrices*. 2008. Disponível em: <https://www.researchgate.net/publication/255652481_Statistical_Texture_Measures_Computed_from_Gray_Level_Cooccurrence_Matrices>. Acesso em: 27 mar 2017. Citado na página 32.

AMORIM, W. P. *Redução de Atributos Utilizando Análise Discriminante com Aplicações na Detecção de Defeitos em Couro Bovino*. Dissertação (Dissertação de Mestrado) — Departamento de Computação e Estatística - Campo Grande/MS, 2009. Citado 4 vezes nas páginas 17, 25, 37 e 45.

BALLARD, D. H.; BROWN, C. M. *Computer Vision*. 1. ed. Nova Jersey: Prentice Hall, 1982. Citado na página 31.

BARIAMIS, D. G. et al. An FPGA-based architecture for real time image feature extraction. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Cambridge, UK: IEEE, 2004. v. 1, p. 801–804. ISSN 1051-4651. Citado na página 44.

BESSA, J. et al. Global location of mobile robots using artificial neural networks in omnidirectional images. v. 13, p. 3405–3414, 10 2015. Citado na página 46.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006. (Information Science and Statistics). ISBN 978-0-387-31073-2. Citado na página 37.

BRYN, L. M. *Página Dinâmica para Aprendizado do Sensoriamento Remoto*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Rio Grande do Sul - UFRGS, 2008. Citado na página 29.

CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, p. 679–698, 1986. Citado na página 54.

CASTAÑON, C. A. B. *Recuperação de imagens por conteúdo através de análise multiresolução por Wavelets*. Dissertação (Tese de Mestrado) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, 2003. Citado na página 55.

CORMEN, T. H. et al. *Introduction to Algorithms*. 3. ed. London, England: The MIT Press Cambridge, Massachusetts, 2009. Citado na página 58.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, Sep 1995. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1007/BF00994018>>. Citado na página 26.

DUAN, K.-B.; KEERTHI, S. S. *Which Is the Best Multiclass SVM Method? An Empirical Study*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. 278–285 p. ISBN 978-3-540-31578-0. Disponível em: <https://doi.org/10.1007/11494683_28>. Citado na página 41.

- ERPEN, L. R. C. *Reconhecimento de Padrões em Imagens por Descritores de Forma*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Rio Grande do Sul - UFRGS, 2004. Citado na página 37.
- FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, v. 27, n. 8, p. 861 – 874, 2006. ISSN 0167-8655. ROC Analysis in Pattern Recognition. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S016786550500303X>>. Citado na página 63.
- FRIEDMAN, J. H. *Another approach to polychotomous classification*. California, USA, 1996. Disponível em: <<http://www-stat.stanford.edu/~jhf/ftp/poly.ps.Z>>. Citado na página 41.
- FURLANETTO, E. L. Mercados nacional e internacional de peles de caprinos e ovinos: uma oportunidade de mercado. *3º Simpósio Internacional sobre Caprinos e Ovinos de Corte - 3º SINCORTE*, 2007. Citado na página 50.
- GONCALVES, A. R. *Máquina de Vetores Suporte*. 2015. Disponível em: <<http://www-users.cs.umn.edu/~andre/arquivos/pdfs/svm.pdf>>. Acesso em: 29 mar 2017. Citado 3 vezes nas páginas 15, 40 e 41.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo: Pearson Prentice Hall, 2010. Citado 5 vezes nas páginas 15, 29, 30, 54 e 55.
- HALL, M. A. *Correlation-based Feature Subset Selection for Machine Learning*. Tese (Doutorado) — University of Waikato, Hamilton, New Zealand, 1998. Citado na página 57.
- HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, p. 786–804, 1979. Citado na página 56.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 1973. Citado 2 vezes nas páginas 32 e 58.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning*. New York: Springer, 2008. Citado na página 41.
- HAYKIN, S. *Neural Networks - A Comprehensive Foundation*. 2. ed. Nova Jersey: Prentice-Hall, 1999. Citado 2 vezes nas páginas 38 e 39.
- IAKOVIDIS, D. K.; MAROULIS, D. E.; BARIAMIS, D. G. FPGA architecture for fast parallel computation of co-occurrence matrices. *Microprocessors and Microsystems*, Elsevier, v. 31, n. 2, p. 160–165, 2007. Citado na página 44.
- JODAS, D. S. et al. Desenvolvimento de um sistema para navegação de robôs móveis por caminhos em plantações. *Interciência & Sociedade*, v. 2, n. 1, p. 27–38, 2013. Disponível em: <<http://hdl.handle.net/11449/122644>>. Citado 2 vezes nas páginas 26 e 43.
- JUNIOR, V. *Fique por dentro do atual mercado do couro no Brasil*. 2015. Revista. Disponível em: <<http://www.risanet.com.br/revista-risa-calcados/revista-risa-maquinas-e-componentes/o-mercado-couro/>>. Acesso em: 22 mar 2017. Citado na página 45.

KRYJAK, T.; KOMORKIEWICZ, M.; GORGON, M. FPGA implementation of real-time head-shoulder detection using local binary patterns, SVM and foreground object detection. In: IEEE. *Design and Architectures for Signal and Image Processing (DASIP), 2012 Conference on*. Karlsruhe, Germany, 2012. p. 1–8. Citado na página 45.

LIN, C. *A support vector machine embedded weed identification system*. Dissertação (Dissertação de Mestrado) — University of Illinois at Urbana-Champaign, 2009. Disponível em: <<https://www.ideals.illinois.edu/handle/2142/14615>>. Citado 2 vezes nas páginas 26 e 44.

LOPEZ-ESTRADA, S.; CUMPLIDO, R. Decision tree based FPGA-architecture for texture sea state classification. In: *2006 IEEE International Conference on Reconfigurable Computing and FPGA's (ReConFig 2006)*. San Luis Potosi, Mexico: IEEE, 2006. p. 1–7. ISSN 2325-6532. Citado na página 44.

LÓPEZ, M. B. et al. Evaluation of real-time LBP computing in multiple architectures. *Journal of Real-Time Image Processing*, Springer, v. 13, n. 2, p. 375–396, 2017. Citado na página 45.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada - RITA*, v. 14, n. 2, 2007. Citado 4 vezes nas páginas 15, 38, 39 e 40.

LYONS, R. G. *Understanding Digital Signal Processing*. 3. ed. Nova Jersey: Prentice-Hall, 2010. 508 p. Citado na página 53.

MÄENPÄÄ, T.; TURPINEN, M.; PIETIKÄINEN, M. Real-time surface inspection by texture. *Real-Time Imaging*, Elsevier, v. 9, n. 5, p. 289–296, 2003. Citado na página 60.

MARINHO, L. B. *Uma nova abordagem de classificação com opção de rejeição para localização e navegação de robôs móveis em mapas topológicos utilizando imagens convencionais e omnidirecionais*. Dissertação (Dissertação de Mestrado) — Instituto Federal de Educação, Ciência e Tecnologia do Ceará - Departamento de Telemática, 2017. Citado 11 vezes nas páginas 15, 26, 31, 38, 47, 50, 51, 52, 61, 73 e 92.

MARINHO, L. B. et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, v. 72, p. 1–17, 2017. ISSN 0957-4174. Citado 3 vezes nas páginas 26, 51 e 55.

MARQUES FILHO, O.; VIEIRA NETO, H. *Processamento Digital de Imagens*. São Paulo: Brasport, 1999. Citado 2 vezes nas páginas 15 e 31.

MARWEDEL, P. *Embedded System Design*. Amsterdam: Springer Netherlands, 2010. ISBN 9789400702561. Citado na página 42.

MATHWORKS. *Texture Analysis Using the Gray-Level Co-Occurrence Matrix (GLCM)*. 2017. Disponível em: <www.mathworks.com>. Acesso em: 27 mar 2017. Citado na página 32.

MEDEIROS, R. O. de. *Manual de Procedimentos Inspeção de Peles do Curtume CV Couros e Peles: Classificação de Wet Blue*. Fortaleza: Curtume CV Couros e Peles, 2007. 11 p. Citado na página 87.

MEHTA, P.; TOMAR, P. An efficient attendance management sytem based on face recognition using matlab and raspberry pi 2. *International Journal of Engineering Technology Science and Research - IJETSR*, v. 3, n. 5, p. 71–78, 2016. Citado 2 vezes nas páginas 26 e 44.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: *Proceedings of 12th International Conference on Pattern Recognition*. Jerusalem, Israel: IEEE, 1994. v. 1, p. 582–585. Citado 2 vezes nas páginas 36 e 56.

OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 7, p. 971–987, 2002. Citado na página 37.

OLIVEIRA, D. C. D.; WEHRMEISTER, M. A. Towards real-time people recognition on aerial imagery using convolutional neural networks. In: *2016 IEEE 19th International Symposium on Real-Time Distributed Computing (ISORC)*. York, UK: IEEE, 2016. p. 27–34. Citado 2 vezes nas páginas 26 e 43.

PERLIN, H. S. L. H. A. Extracting human attributes using a convolutional neural network approach. *Pattern Recognition Letters*, p. 250–259, 2015. Citado na página 63.

PIETIKÄINEN, M. Local Binary Patterns. *Scholarpedia*, v. 5, n. 3, p. 9775, 2010. Revision #183551. Citado na página 35.

PIETIKÄINEN, M. et al. *Computer Vision Using Local Binary Patterns*. London: Springer, 2011. v. 40. (Computational Imaging and Vision, v. 40). ISBN 978-0-85729-748-8. Citado 2 vezes nas páginas 37 e 44.

PIETIKÄINEN, M. et al. Local binary patterns for still images. In: *Computer vision using local binary patterns*. London: Springer London, 2011. p. 13–47. Citado na página 60.

PIMENTEL, A. D. Exploring exploration: A tutorial introduction to embedded systems design space exploration. *IEEE Design Test*, v. 34, n. 1, p. 77–90, Feb 2017. ISSN 2168-2356. Citado na página 42.

PLATT, J. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. [S.l.], 1998. 21 p. Disponível em: <<https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/>>. Citado na página 45.

POWERS, D. M. W. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, v. 2, p. 37 – 63, 2011. ISSN 2229-3981. Disponível em: <<http://www.bioinfo.in/contents.php?id=51>>. Citado na página 63.

ROCHA NETO, A. R. da. *SINPATCO II: Novas Estratégias de Aprendizado de Máquinas para Classificação de Patologias da Coluna Vertebral*. Dissertação (Tese de Doutorado) — Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará - UFC, 2011. Citado 4 vezes nas páginas 17, 39, 40 e 41.

ROCHA NETO, A. R. da; BARRETO, G. de A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions*, v. 7, n. 4, p. 487–496, Aug 2009. ISSN 1548-0992. Citado na página 62.

SANTOS FILHO, E. Q. dos. *Um Sistema de Visão Computacional para Classificação da Qualidade do Couro Caprino*. Dissertação (Dissertação de Mestrado) — Departamento de Engenharia de Teleinformática - UFC, 2013. Citado 9 vezes nas páginas 17, 25, 26, 46, 50, 51, 61, 72 e 92.

SETHUPATHY, J.; Veni S. Opencv based disease identification of mango leaves. *International Journal of Engineering Technology Science and Research - IJETSR*, v. 8, n. 5, p. 1990–1998, 2016. Citado 2 vezes nas páginas 26 e 44.

SHI, G.; CHAN, C. S.; LI, W. J. Mobile human airbag system for fall protection using mems sensors and embedded SVM classifier. *IEEE Sensors Journal*, v. 9, p. 495–503, 2009. Citado 2 vezes nas páginas 26 e 43.

SIELER, L.; TANOUGAST, C.; BOURIDANE, A. A scalable and embedded FPGA architecture for efficient computation of grey level co-occurrence matrices and haralick textures features. *Microprocessors and Microsystems*, Elsevier, v. 34, n. 1, p. 14–24, 2010. Citado na página 44.

SILVIA, C. Y. V. W. da. *Extração de características de imagens médicas utilizando wavelets para mineração de imagens e auxílio ao diagnóstico*. Dissertação (Tese de Mestrado) — Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, 2007. Citado 2 vezes nas páginas 25 e 55.

SMOLA, A. J. et al. *Advances in Large Margin Classifiers*. Cambridge: Massachusetts Institute of Technology, 2000. ISBN 0-262-19448-1. Citado na página 41.

SOH, L.-K.; TSATSOULIS, C. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, v. 37, n. 2, 1999. Citado 2 vezes nas páginas 32 e 58.

TAHIR, M. A. et al. Accelerating the computation of GLCM and haralick texture features on reconfigurable hardware. In: IEEE. *Image Processing, 2004. ICIP'04. 2004 International Conference on*. Singapore, 2004. v. 5, p. 2857–2860. Citado na página 44.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition, Fourth Edition*. Cambridge, Massachusetts: Academic Press, 2008. ISBN 9781597492720. Citado na página 37.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=211359>>. Citado na página 39.

VIEIRA, R. G. *Processo de Prototipação de um Classificador Neural Embarcado para Detecção de Falhas em Motores de Indução Trifásicos*. Dissertação (Dissertação de Mestrado) — Instituto Federal de Educação, Ciências e Tecnologia do Ceará - Departamento de Telemática, 2016. Citado na página 27.

WANG, H. et al. Multilevel binomial logistic prediction model for malignant pulmonary nodules based on texture features of ct image. *European Journal of Radiology*, v. 74, p. 124–129, 2010. Citado 2 vezes nas páginas 32 e 58.

WOLF, W. *Computers as Components: Principles of Embedded Computing System Design*. 4. ed. Burlington, Massachusetts: Morgan Kaufmann, 2016. ISBN 9780128053874. Citado 2 vezes nas páginas 25 e 42.