



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
PROGRAMA DE MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

JÔNATAS HOLANDA NOGUEIRA DE AQUINO

**EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS PARA CLASSIFICAÇÃO DA
QUALIDADE DE COURO CAPRINO USANDO PADRÃO BINÁRIO LOCAL**

FORTALEZA

2017

JÔNATAS HOLANDA NOGUEIRA DE AQUINO

EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS PARA CLASSIFICAÇÃO DA
QUALIDADE DE COURO CAPRINO USANDO PADRÃO BINÁRIO LOCAL

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Mestrado e Doutorado em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. João Fernando Lima Alcântara

Co-Orientador: Prof. Dr. Francesco Corona

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A669e Aquino, Jônatas Holanda Nogueira de.

Extração de características de imagens para classificação da qualidade de couro caprino usando padrão binário local / Jônatas Holanda Nogueira de Aquino. – 2017.
63 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.

Orientação: Prof. Dr. João Fernando Lima Alcântara.

Coorientação: Prof. Dr. Francesco Corona.

1. Padrão Binário Local. 2. Classificação de Textura. 3. Classificação de Couro. 4. Máquina de Vetor de Suporte. 5. K Vizinhos Mais Próximos. I. Título.

CDD 005

JÔNATAS HOLANDA NOGUEIRA DE AQUINO

EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS PARA CLASSIFICAÇÃO DA
QUALIDADE DE COURO CAPRINO USANDO PADRÃO BINÁRIO LOCAL

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Mestrado e Doutorado em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. João Fernando Lima Alcântara
(Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Francesco Corona (Co-Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará (UFC)

Prof. Dr. Ajalmar Rêgo da Rocha Neto
Instituto Federal de Educação, Ciência e Tecnologia
do Ceará (IFCE)

A Deus, aos meus pais Erivan e Sueli e a minha
irmã Valéria.

AGRADECIMENTOS

Agradeço a Deus.

Agradeço à minha Família por todo apoio.

Agradeço aos professores e orientadores que fizeram parte desta etapa de minha formação, bem como aos colegas de laboratório e amigos pelo apoio durante a realização deste trabalho.

Agradeço, também, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento do trabalho.

"Este é o meu mandamento: amai-vos uns aos outros, assim como eu vos amei."

(João 15:12)

RESUMO

Muitos problemas de visão computacional podem ser solucionados por meio de métodos de reconhecimento de textura. Problemas como análise de imagens de satélite, inspeção industrial, diagnóstico médico por imagens, reconhecimento de faces exemplificam aplicações em que podem ser utilizadas técnicas de classificação de textura. Dentre os diferentes métodos de extração de característica está o *Local Binary Pattern* (LBP) ou Padrão Binário Local, que tem se destacado nos últimos anos e possui diversas variantes direcionadas à solução de diferentes tipos de problemas.

No contexto do nordeste brasileiro, aplicações de classificação de textura encontram potencial utilidade na indústria de couro caprino, onde a classificação da qualidade das peças de couro é feita manualmente e de maneira subjetiva. Neste cenário, a automatização da classificação pode ajudar a padronizar o processo de identificação da qualidade de couro, tornando a tarefa mais ágil e objetiva.

Neste sentido, este trabalho apresenta os resultados da aplicação dos métodos mais tradicionais de Padrões Binários Locais para extração de características de imagens de couro caprino, verificando a viabilidade para a classificação de sete classes de qualidade. Na proposta, diferentes parâmetros de extração são testados e combinações de extratores LBP são usadas com o objetivo de se obter a melhor configuração de extração necessária para a solução do problema. A análise inclui ainda uma comparação entre o KNN e o SVM para a realização da classificação.

Palavras-chave: Padrão Binário Local. LBP. Classificação de Textura. Textura. Máquina de Vetor de Suporte. SVM. KNN. Classificação de Couro

ABSTRACT

Many computer vision problems find solution in texture recognition methods. Tasks in different areas such as satellite image analysis, industrial inspection, medical image analysis and facial recognition exemplify applications where texture recognition techniques can be used. Local Binary Pattern (LBP) is among a list of different methods for texture feature extraction and has called attention of many academics in the last years and many different variations has been develop to address different kinds of real-world problems.

In the Brazilian northeast, texture classification has special potential in the leather processing industry for leather quality classification, which is currently done manually and subjectively. In this scenario, the automation of the quality classification can help to improve the labeling process and at the same time to make process faster and objective.

In this sense, this dissertation presents the results of applying the most traditional local binary pattern methods for feature extraction in caprine leather, as well as observing the feasibility for classify seven different types of quality classes. In this work, many different parameters for extraction are tested and LBP combinations are made in order to reach a better feature representation capable of solving the problem. The analysis also includes a comparison between KNN and SVM.

Keywords: Local Binary Pattern. LBP. Texture Classification. Support Vector Machine. SVM. KNN. Leather Classification.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sequência de Aprendizagem Automática	19
Figura 2 – Visualização da região de decisão para $K=1$	22
Figura 3 – Equação do hiperplano	23
Figura 4 – Distância entre hiperplanos	24
Figura 5 – SVM maximização da margem	25
Figura 6 – Múltiplas Classes	29
Figura 7 – Mesa de Aquisição das Imagens	31
Figura 8 – Imagem Capturada	32
Figura 9 – Exemplo de defeitos em couro bovino. Imagem superior origem do defeito. Imagem inferior o resultado no estado <i>Wet-Blue</i> . a) defeito sarna, b) defeito carrapato, c) defeito risco.	33
Figura 10 – Regiões do couro	34
Figura 11 – Exemplo de texturas, com a extração de suas respectivas imagens e histograma LBP	37
Figura 12 – Representação geométrica do formalismo da Extração LBP	38
Figura 13 – Ilustração da Extração LBP	39
Figura 14 – Padrões uniformes para 8 pontos.	40
Figura 15 – Determinando o número de transições de 0 para 1 ou 1 para zero. Exemplificando a aplicação da Equação (4.2) para uma sequência de 8 bits. A sequencia foi espaçada a cada 4 bits para facilitar a visualização.	41
Figura 16 – Exemplo de padrões invariantes à rotação. A linha superior é equivalente a linha inferior.	43
Figura 17 – Comparação entre LBP U2 e LBP RIU2 (Textura Artificial 200x200)	45
Figura 18 – Entendendo os padrões da textura artificial da Figura 17. Cada deslocamento representa os tipos de LBP(8,1) possíveis na textura de listras verticais. Note que quando o pixel central é zero, todos os pixels vizinhos são maiores ou iguais a ele.	46
Figura 19 – Diagrama da Extração com Combinação Simples	51
Figura 20 – Diagrama detalhado da região pontilhada da figura 19	52
Figura 21 – Diagrama da Extração com Combinação Dupla	54
Figura 22 – Diagrama da Extração com Combinação Tripla	57

LISTA DE TABELAS

Tabela 2 – Parâmetros para classificação manual de couro	34
Tabela 3 – Tamanho do vetor LBP sem combinação	49
Tabela 4 – Tamanho do vetor da combinação de 2 LBP	49
Tabela 5 – Tamanho do vetor da combinação de 3 LBP	50
Tabela 6 – Acurácia sem combinar LBP	52
Tabela 7 – Matriz de Confusão Operador Simples RIU2 com KNN	53
Tabela 8 – Matriz de Confusão Operador Simples RIU2 com SVM	53
Tabela 9 – Melhor Acurácia média Combinando 2 Operadores LBP	55
Tabela 10 – Matriz de Confusão Combinando 2 Operadores RIU2 com KNN	56
Tabela 11 – Matriz de Confusão Combinando 2 Operadores RIU2 com SVM	56
Tabela 12 – Melhor Acurácia média Combinando 3 Operadores LBP	57
Tabela 13 – Matriz de Confusão Combinando 3 Operadores RIU2 com KNN	57
Tabela 14 – Matriz de Confusão Combinando 3 Operadores RIU2 com SVM	58
Tabela 15 – Melhor Acurácia média sem combinar LBP - Detalhada	62
Tabela 16 – Melhor Acurácia média Combinando 2 Operadores LBP - Detalhada	62
Tabela 17 – Melhor Acurácia média Combinando 3 Operadores LBP - Detalhada	62

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo do Classificador KNN	21
--	----

LISTA DE ABREVIATURAS E SIGLAS

$LBP_{P,R}^{riu2}$	padrão binário local invariante à rotação e uniforme
$LBP_{P,R}^i$	padrão binário local invariante à rotação
$LBP_{P,R}^{u2}$	padrão binário local uniforme
GLCM	matrizes de co-ocorrência em níveis de cinza
HAAR	transformada wavelet de HAAR
KNN	k-vizinhos mais próximos
LBP	padrão binário local
NMF	fatoração de matrizes não-negativas
PCA	análise das componentes principais
SVM	máquinas de vetor de suporte
VAR	variância das colunas

LISTA DE SÍMBOLOS

P	Número de pontos da vizinhança do LBP
R	Valor do raio LBP
g_c	Nível de cinza do pixel central
g_p	Nível de cinza do pizel da vizinhança LBP
$s(x)$	Função sinal
$U(x)$	Operador uniforme
K	Núcleo da máquina de vetor de suporte
\mathbf{w}	Vetor de pesos
\mathbf{x}	Vetor x
y	Vetor rótulos
D	Distância entre dois planos
α	Multiplicadores de Lagrange
β	Multiplicadores de Lagrange
L	Lagrangiano

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	16
1.1.1	<i>Objetivos específicos</i>	16
2	APRENDIZAGEM DE MÁQUINA	18
2.1	Classificação	19
2.1.1	<i>K-vizinhos mais próximos (KNN)</i>	21
2.1.2	<i>Máquinas de Vetor de Suporte (SVM)</i>	22
2.1.3	<i>Derivando a equação da Distância entre dois hiperplanos</i>	22
2.1.4	<i>SVM Formulação Primal</i>	25
2.1.5	<i>SVM Formulação Dual</i>	26
2.1.6	<i>SVM para Margens Suaves</i>	28
2.1.7	<i>SVM Kernel-Trick</i>	28
2.2	Classificação com Múltiplas Classes	29
3	AQUISIÇÃO DE DADOS	31
4	EXTRAÇÃO DE CARACTERÍSTICAS USANDO LBP	36
4.1	Textura em Processamento de Imagens	36
4.2	Padrão Binário Local (LBP)	37
4.3	Método LBP Uniforme	40
4.4	Método LBP Invariante à Rotação	42
4.5	Método LBP Uniforme e Invariante à Rotação	44
5	CLASSIFICAÇÃO DA QUALIDADE DE COURO CAPRINO UTILIZANDO LBP	47
5.1	Representação do Vetor de características	48
5.2	Extração Simples com 1 Operador LBP	50
5.3	Extração Multi-resolução com 2 Operadores LBP	53
5.4	Extração Multi-resolução com 3 Operadores LBP	55
6	CONCLUSÕES E TRABALHOS FUTUROS	59
	REFERÊNCIAS	60
7	RESULTADOS COM CONFIGURAÇÃO DOS PARÂMETROS LBP	62
8	TEMPO DE PROCESSAMENTO	63

8.1	Tempo do classificador simples	63
8.2	Tempo do classificador com 2 combinações	63
8.3	Tempo do Classificador com 3 combinações	63

1 INTRODUÇÃO

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

A classificação de texturas pode ser realizada utilizando diferentes métodos de processamento de imagem. Uma das primeiras técnicas desenvolvidas utiliza Matrizes de Dependência Espacial em Tons de Cinza (*Gray-Tone Spatial-Dependence Matrices*), onde a partir dessa matriz se extrai um conjunto de características tais como contraste, correlação, momento angular, entropia dentre outras medições estatísticas que visam a representação da textura (HARALICK *et al.*, 1973).

Mais recentemente o uso de Padrões Binários Locais, LBP, proposto por Ojala (OJALA *et al.*, 2002), vem se destacando em vários ramos de processamento de imagens por sua capacidade de representar texturas e também por sua formalização permitir modificações mantendo a mesma estrutura lógica de extração. Atualmente, existem diversas variantes do método LBP que se aplicam não só na identificação de texturas mas também no reconhecimento de faces e segmentação de imagens (LIU *et al.*, 2017).

Na extração de características de textura, dependendo da aplicação, alguns aspectos do problema deve ser levado em consideração, por exemplo, características como a variação da luminosidade de imagem, mudanças na escala, na rotação da imagem e tipo de ruído podem influenciar a extração de atributos da textura e alterar a performance do classificador.

Aplicações de classificação de textura no contexto brasileiro, mais especificamente no nordestino, encontram potencial utilidade na indústria de couro caprino, onde a classificação da qualidade é feita manualmente e de maneira subjetiva. Nesse cenário, a automatização da classificação pode agilizar e padronizar o processo de identificação da qualidade do couro.

No trabalho de Santos Filho, uma abordagem é apresentada para o problema da classificação da qualidade do couro na qual em sua dissertação são realizados testes em um banco de imagens utilizando cinco métodos, variância das colunas (VAR), matrizes de co-ocorrência em níveis de cinza (GLCM), análise das componentes principais (PCA), fatoração de matrizes não-negativas (NMF) e transformada wavelet de HAAR (HAAR). Porém, as imagens utilizadas

foram convertidas do tamanho original de 2364x2448 para 40x40, 60x60, 80x80 e 100x100 (SANTOS FILHO, 2013). Além disso, as classes de imagens foram binarizadas reduzindo de 7 tipos de qualidade para apenas 2 tipos, onde as cinco primeiras classes foram denominadas classe superior, e a classe 6 e 7 como classe inferior, referenciando as peças de couro com maior ou menor valor de mercado.

Apesar dos esforços para a classificação de textura, o trabalho ainda possui diversos aspectos que podem ser explorados e melhorados, como a verificação da eficiência do LBP na extração de atributos, tema deste trabalho.

Dado o que foi exposto, este trabalho se propõe a aplicar o método LBP na extração de atributos das imagens de couro caprino e avaliar a capacidade do método de identificar diferentes tipos de qualidade de couro caprino. Para isso, será feita a análise do desempenho do método comparando e combinando as variantes mais tradicionais para então escolher a melhor maneira de extrair características relevantes para aplicação. Além disso, pretende-se aplicar algoritmos de aprendizagem automática tendo ao final uma aplicação prática de visão computacional e inteligência artificial.

1.1 Objetivos

A pesquisa tem como principal objetivo classificar imagens de couro caprino, diferenciando as peças de melhor qualidade das peças de qualidade inferior. Pretende-se discriminar um total de sete classes de imagens utilizando o método Padrão Binário Local (LBP) como método de extração de atributos.

1.1.1 *Objetivos específicos*

Fazer a aplicação de diferentes métodos LBP em uma base de dados de 350 imagens de couro caprino, e avaliar o efeito da variação dos parâmetros de extração na discriminação de cada classe de qualidade de couro. Serão aplicados os extratores de características LBP uniforme ($LBP_{P,R}^{u2}$), LBP invariante à rotação ($LBP_{P,R}^{ri}$) e o LBP uniforme e invariante à rotação ($LBP_{P,R}^{riu2}$). Para cada um desses extratores será feita a variação dos valores de P e R , referentes ao número de pontos da vizinhança do LBP e ao valor do raio. Cada extrator LBP aplicado sobre a base de imagens gera uma nova base de dados para cada par (P,R) . Cada base de dados gerada será dividida em treino e teste, e usada para treinar um classificador KNN e um SVM.

Cada base de dados terá uma acurácia que está relacionada com os parâmetros (P,R) para extrair as características. A base que treinar o melhor classificado será usada como referência para a escolha do melhor extrator LBP e seus parâmetros de extração (P,R).

2 APRENDIZAGEM DE MÁQUINA

Existem várias definições para aprendizagem de máquina, porém podemos de maneira simplificada dizer que a aprendizagem de máquina consiste em programar o computador para que ele seja capaz de realizar uma determinada tarefa sobre novos dados tendo como base um treinamento prévio, feito a partir de um conjunto de dados já conhecido. Essa tarefa pode ser de classificação, regressão ou agrupamento.

Existem diversos métodos para fazer com que a máquina reconheça padrões os quais podem ser divididos em três grandes grupos a saber: supervisionados, não supervisionados e semi-supervisionado, sendo que neste trabalho foram aplicados apenas métodos supervisionados.

O aprendizado supervisionado reúne os métodos que utilizam uma base de dados previamente rotulada, ou seja, a partir do treinamento feito com padrões observáveis e devidamente classificados. Como exemplo desse tipo de método temos o SVM e as Redes Neurais Artificiais e o *Naive Bayes*.

No aprendizado não supervisionado estão inclusos os métodos baseado em agrupamento, tal como o k-médias (*k-means*). Nesse tipo de aprendizado não temos conhecimento sobre o significado dos dados, ou seja, não se tem uma classificação prévia, em outras palavras, não temos os rótulos. Sem a rotulação dos dados, é feito o agrupamento de dados a partir de uma métrica de distância para inferir alguma relação de similaridade entre os dados e, assim, extrair algum conhecimento inicial. Um exemplo clássico é a relação encontrada entre os compradores de cerveja e fraudas em uma base de dados de um supermercado, assim, a clusterização, mesmo sem informação de rótulos, ajudou a organização dos produtos baseados na correlação entre as duas compras.

Já o aprendizado semi-supervisionado é útil quando temos uma base de dados parcialmente rotulada, ou seja, apesar de termos alguma informação sobre os dados, ainda assim existem muitos dados desconhecidos. Assim, conseguir rotular esses dados a partir da informação já existente, resultaria em uma modelagem mais completa. Por exemplo, em algumas situações práticas o custo de rotular um determinado volume de dados é alto, muitas vezes devido ao tempo ou ao número de especialistas necessários para a realização da tarefa. Nestes casos, pode-se seguir duas estratégias. A primeira consiste em rotular a base até onde for viável e então descartar o restante dos dados não rotulado. Na segunda abordagem também se rotula os dados até onde for viável, porém, ao invés de descartar os dados desconhecidos aplicam-se técnicas de aprendizado semi-supervisionado. Assim o restante da base é rotulada automaticamente,

melhorando a estimativa do modelo.

Vale ainda ressaltar que os classificadores podem ser divididos entre discriminativos e generativos. Os classificadores discriminativos, como a regressão logística, geram uma superfície de decisão mapeando diretamente a entrada x aos rótulos y . Já os classificadores generativos, como o *Naive Bayes*, modelam a função de probabilidade para então decidir o rótulo mais provável (NG; JORDAN, 2002).

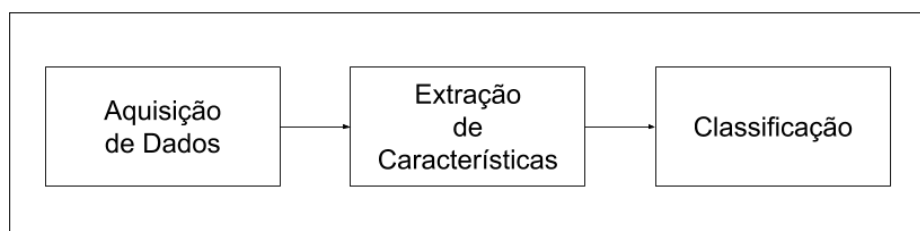
Este trabalho se caracteriza como aprendizagem supervisionada, visto que possuímos informação sobre os rótulos das imagem. Na Seção 2.1 iremos abordar de maneira resumida o processo de classificação, desde a aquisição das imagens, passando pela extração de atributos e os métodos classificação utilizados, no caso o k-vizinhos mais próximos (KNN) e as máquinas de vetor de suporte (SVM). O capítulo 3 aprofundará os aspectos relacionados a aquisição das imagens. Já o capítulo 4 será dedicado aos métodos de extração de atributos.

2.1 Classificação

O processo de classificação de padrões pode ser dividida em três etapas. A etapa de aquisição de dados, a etapa de extração de características e a classificação em si. A Figura 1 apresenta o fluxo comum dessas três etapas no processo de aprendizagem automática.

A primeira etapa consiste na aquisição dos dados do mundo real, que dependendo da aplicação podem ser fotos de satélite, imagem de radiografia, tabela de preços, histórico de vendas de uma empresa, dentre outras que de alguma forma foram inseridas no computador seja manualmente, ou automaticamente por meio de sensores.

Figura 1 – Sequência de Aprendizagem Automática



Fonte – Adaptado de (GOMES, 2006)

Mesmo depois de digitalizados, o volume de dados pode ser grande, podendo conter mais informação do que o necessário para a aplicação alvo. Neste caso, se a etapa de classificação for aplicada diretamente sobre os dados brutos a aplicação pode se tornar inviável. O tempo

de processamento pode ser demasiadamente longo, ou a classificação pode não ser satisfatória. Para contornar esses problemas citados e se obter um reconhecimento de padrão mais eficiente, utilizamos a etapa de extração de características, obtendo assim dados mais relevantes.

A etapa de extração de características filtra a parte mais importante dos dados para a etapa de classificação (LIU; MOTODA, 1998), o que acaba tornando essa tarefa dependente do tipo da aplicação. Por exemplo, para identificar faces pode ser útil se obter informação sobre a geometria da face como distância relativa entre olhos, nariz e boca (BRUNELLI; POGGIO, 1993). Em aplicações médicas, como diagnóstico de patologias na coluna vertebral, as características extraídas pelo médico especialista que irá identificar um conjunto de angulações das vértebras e para cada conjunto de atributos, dará um rótulo, no caso, um diagnóstico (ROCHA NETO *et al.*, 2011). Na classificação de *spam* em caixa de correio eletrônico, a contagem das palavras mais frequentes de *e-mail* em mensagens indesejadas poderá ser extraídas a partir do processamento do texto. Em resumo, o tipo de informação a ser extraída é uma decisão importante no processo de classificação.

No caso deste trabalho, estamos interessados em obter informação de textura das imagens para conseguir identificar o tipo de qualidade da peça de couro. Neste sentido, precisamos extrair informação de textura das imagens, e o método para a extração escolhido foi o Padrão Binário Local (*Local Binary Patterns*, LBP) que realiza uma operação matemática sobre a imagem extraindo um vetor com informação sobre textura e que será explicada em mais detalhes no Capítulo 4.

Por fim, temos a etapa de classificação. Nessa etapa vários tipos de métodos podem ser aplicados, podendo, cada um, apresentar desempenho diferentes, para cada tipo de aplicação. Dentre os métodos mais comuns para aprendizagem supervisionada podemos citar as Máquinas de Vetor de Suporte, as Redes Neurais, o KNN (*K-Nearest Neighbour*), e Métodos Estatísticos como *Naive Bayes*. Existem ainda as variações de cada classificador que visam melhorar o desempenho para algum tipo aplicações com características específicas, o que indica que a escolha do classificador também poderá depender da aplicação.

Assim, é importante testar mais de um tipo de classificador. Apesar do foco deste trabalho ser a extração de características de textura, apresentamos também uma comparação dos métodos KNN e SVM de *Kernel* Polinomial. A seguir iremos descrever um pouco mais sobre cada um desses métodos e ao final mostraremos os problemas comuns relacionado a classificação com múltiplas classes.

2.1.1 K-vizinhos mais próximos (KNN)

O K-vizinhos mais próximos, ou KNN é um método baseado em instâncias e não possuindo uma etapa de treinamento para o calculo de uma função de hipótese. O método apenas armazena as amostras de treinamento e a classificação é feita de forma que para cada nova entrada, verifica-se quem são os K-vizinhos mais próximos e atribui a essa nova entrada o rótulo da maioria.

A distância euclidiana é normalmente usada para determinar as instâncias mais próximas do vetor de entrada e pode ser calculada da seguinte forma. Dado dois vetores \mathbf{x}_i e \mathbf{x}_j

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}. \quad (2.1)$$

Abaixo um algoritmo para a implementação do KNN.

Algoritmo 1: Algoritmo do Classificador KNN

Entrada: Uma instancia \mathbf{x}_q , um inteiro K, e um conjunto de instâncias $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

Crie um a lista BASE representando a base de treino;

Crie um mapa de distancia DIST<v,distancia>;

Crie uma lista V que armazenará os k primeiro vetores para posterior decisão da classificação.

início

para i de 1 até n **faça**

 BASE.add(< \mathbf{x}_i, y_i >);

 DIST.add($\mathbf{x}_q, d(\mathbf{x}_q, \mathbf{x}_i)$);

fim

Ordene o mapa DIST de acordo com a distancia.

Adicione em V os k primeiros vetores de DIST.

Retorne:

$$y_k \leftarrow \underset{v \in L}{\operatorname{argmax}} \{ \sum_{i=1}^K \delta(v, V(i)) \}$$

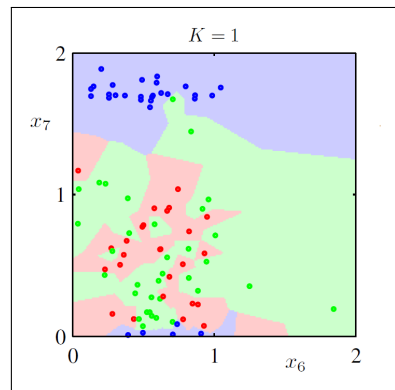
onde $\delta(a, b) = 1$ se $a = b$, caso contrário $\delta(a, b) = 0$. Além disso L é o conjunto dos

Rótulos;

fim

O Algoritmo 1 de classificação do KNN, baseado em (MITCHELL, 1997), retorna o rótulo mais comum dentre os k dos exemplos de treinamento que estão mais próximos do exemplo que se deseja estimar.

Figura 2 – Visualização da região de decisão para $K=1$



Fonte – (BISHOP, 2006)

Apesar de não gerar explicitamente uma função que represente a região de decisão, ou uma de hipótese, podemos imaginar a região de decisão do KNN para um determinado K como sendo uma espécie de diagrama de Voronoi (MITCHELL, 1997). Como pode ser visto na Figura 2, o espaço de decisão para $K=1$, onde cada ponto está envolvido por um poliedro que varia de acordo com os pontos da vizinhança. Note que os poliedros contíguos associados a um mesmo rótulo acabam formando uma região maior.

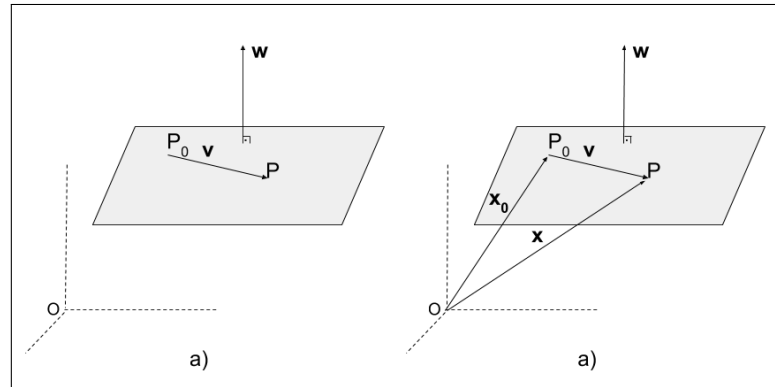
2.1.2 Máquinas de Vetor de Suporte (SVM)

As Máquinas de Vetor de Suporte têm sido bem sucedidas na classificação por conseguir maximizar a margem de separação entre as classes da região de decisão com relação aos dados de treinamento (candidatos à vetores de suporte), resultando em um modelo mais generalista melhorando o desempenho de classificação. A Figura 5b) ilustra um conjunto de dados binários com a superfície de decisão definida pela linha sólida e as margens pelas linhas pontilhadas. A intuição é que quanto mais próximo da superfície de decisão, menos certeza temos sobre a classificação, assim, quanto maior for a margem, melhor desempenho terá o modelo.

2.1.3 Derivando a equação da Distância entre dois hiperplanos

Para uma apresentação mais amigável da formulação do SVM, será apresentado a equação do hiperplano e em seguida a equação da distância entre dois hiperplanos de acordo com a sequencia didática baseada em Statnikov (STATNIKOV *et al.*, 2011). As figuras estarão no \mathcal{R}^3 , porém, a explicação pode ser generalizada para dimensões maiores que 3.

Figura 3 – Equação do hiperplano



Fonte – O autor, baseado em (STATNIKOV *et al.*, 2011)

Imagine um hiperplano em \mathfrak{R}^3 cujo vetor normal é dado por \mathbf{w} de acordo com a Figura 3a) e que passa pelo ponto P_0 . Tomando um ponto qualquer P , o hiperplano pode ser descrito pelo produto interno dos vetores \mathbf{w} e o vetor formado pelo seguimento $P_0P = \mathbf{v}$. Como o vetor \mathbf{v} é perpendicular ao vetor normal \mathbf{w} , temos:

$$\mathbf{w}^T \mathbf{v} = 0$$

Em um segundo passo, descrito pela Figura 3b) imagine que os pontos P_0 e P possam ser representados por vetores com origem no ponto O , a origem do sistema de coordenadas. Veja que o vetor $\mathbf{x} = \mathbf{x}_0 + \mathbf{v}$, ou seja, $\mathbf{v} = \mathbf{x} - \mathbf{x}_0$. Podemos definir o plano agora por:

$$\mathbf{w}^T (\mathbf{x} - \mathbf{x}_0) = 0$$

Distribuindo temos: $\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{x}_0 = 0$, e portanto, podemos considerar $b = -\mathbf{w}^T \mathbf{x}_0$ uma vez que \mathbf{w} e \mathbf{x}_0 são constantes que definem o hiperplano. Assim:

$$\mathbf{w}^T \mathbf{x} + b = 0 \tag{2.2}$$

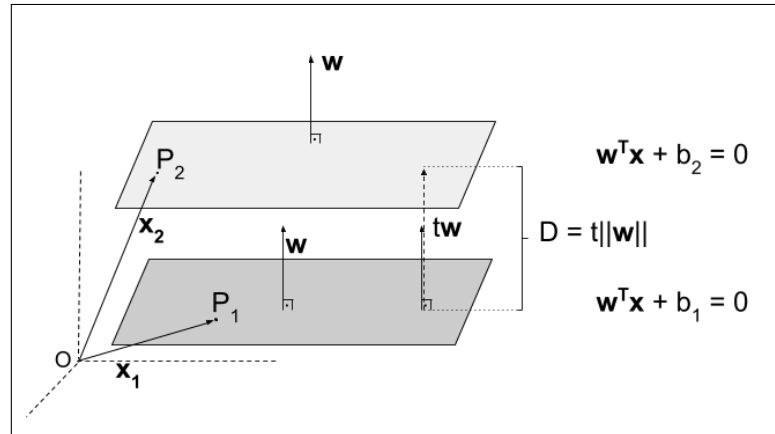
Para a definição da equação da distância, conforme ilustrado pela Figura 4, tome dois hiperplanos paralelos no \mathfrak{R}^3 , com normal \mathbf{w} , um contendo o ponto P_1 e outro contendo o ponto P_2 . Podemos representar os pontos P_1 e P_2 por vetores assim como foi feito na Figura 3, bem como qualquer ponto do espaço por vetores com origem em O . Note que os vetores \mathbf{x} que satisfazem a equação do hiperplano 2 podem ser definidos em termos dos vetores do plano 1 adicionados da distância entre eles, definida por $D = t \|\mathbf{w}\|$.

Assim temos as seguintes equações dos hiperplanos:

$$\mathbf{w}^T \mathbf{x} + b_1 = 0 \text{ (Hiperplano 1)}$$

$$\mathbf{w}^T \mathbf{x} + b_2 = 0 \text{ (Hiperplano 2)}$$

Figura 4 – Distância entre hiperplanos



Fonte – O autor, baseado em (STATNIKOV *et al.*, 2011)

Em outras palavras, a equação do hiperplano 2 pode ser definida em termos do hiperplano 1. Veja que cada vetor que satisfazem o hiperplano 2 pode ser reescrito na forma de vetores do hiperplano 1 adicionados do vetor $t\mathbf{w}$ que representa a distância entre os dois planos. Por conveniência, iremos representar um vetor qualquer do hiperplano 1 pelo vetor \mathbf{x}_1 e um vetor qualquer do hiperplano 2 está sendo representado por \mathbf{x}_2 .

$$\mathbf{x}_2 = \mathbf{x}_1 + t\mathbf{w}$$

Agora podemos reescrever o hiperplano 2 da seguinte forma:

$$\mathbf{w}^T \mathbf{x}_2 + b_2 = 0$$

$$\mathbf{w}^T (\mathbf{x}_1 + t\mathbf{w}) + b_2 = 0$$

$$\mathbf{w}^T \mathbf{x}_1 + t\mathbf{w}^T \mathbf{w} + b_2 = 0$$

Artifício matemático soma e subtrai b_1

$$\mathbf{w}^T \mathbf{x}_1 + b_1 - b_1 + t\mathbf{w}^T \mathbf{w} + b_2 = 0$$

$$(\mathbf{w}^T \mathbf{x}_1 + b_1) - b_1 + t\mathbf{w}^T \mathbf{w} + b_2 = 0$$

Reescrevendo $\mathbf{w}^T \mathbf{w} = \|\mathbf{w}\|^2$

$$(\mathbf{w}^T \mathbf{x}_1 + b_1) - b_1 + t\|\mathbf{w}\|^2 + b_2 = 0$$

Sabemos pela equação do hiperplano 1 que $\mathbf{w}^T \mathbf{x}_1 + b_1 = 0$

$$-b_1 + t\|\mathbf{w}\|^2 + b_2 = 0$$

Assim temos que o valor de t é igual a:

$$t = (b_1 - b_2) / \|\mathbf{w}\|^2$$

Como a distância é dada por $D = t\|\mathbf{w}\|$, se substituirmos o valor de t teremos a

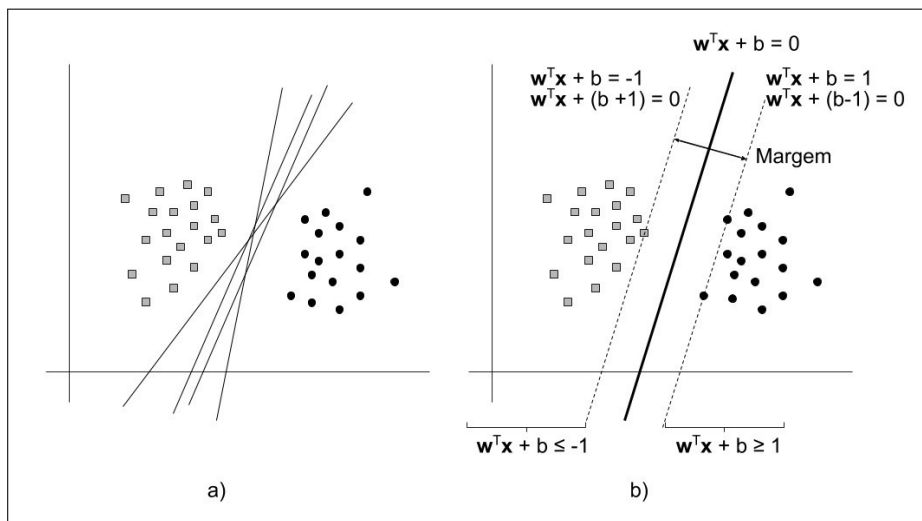
seguinte equação da distância entre dois hiperplanos:

$$D = \frac{|b_1 - b_2|}{\|\mathbf{w}\|} \quad (2.3)$$

2.1.4 SVM Formulação Primal

Dado um conjunto de dados linearmente separáveis, podemos encontrar infinitos hiperplanos de separação de modo a obter um modelo para a classificação dos dados. No entanto, estamos interessados em encontrar o hiperplano com maior margem de separação entre as duas classes.

Figura 5 – SVM maximização da margem



Fonte – O autor, baseado em (STATNIKOV *et al.*, 2011)

A distância entre os dois hiperplanos na Figura 5:

$$\mathbf{w}^T \mathbf{x} + b = -1 \text{ e}$$

$$\mathbf{w}^T \mathbf{x} + b = +1$$

também equivalente a

$$\mathbf{w}^T \mathbf{x} + (b + 1) = 0$$

$$\mathbf{w}^T \mathbf{x} + (b - 1) = 0$$

De acordo com a distância entre dois hiperplanos dada pela Equação (2.3) temos que

$$D = (b_1 - b_2) / \|\mathbf{w}\|, \text{ No caso, } b_1 = (b + 1) \text{ e } b_2 = (b - 1).$$

Assim, a margem do é dada por:

$$D = 2 / \|\mathbf{w}\|$$

Como queremos maximizar a margem, devemos minimizar $\|\mathbf{w}\|$, que equivale a minimizar $\|\mathbf{w}\|^2$ e por conveniência matemática iremos multiplicar por 1/2.

$$\min\{\frac{1}{2}\|\mathbf{w}\|^2\}$$

Precisaremos, também, impor as seguintes restrições:

$$\mathbf{w}^T \mathbf{x} + b \leq -1 \text{ se } y_i = -1$$

$$\mathbf{w}^T \mathbf{x} + b \geq +1 \text{ se } y_i = +1$$

equivalente à

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \text{ para todo } i=1,\dots,N$$

ou

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \text{ para todo } i=1,\dots,N$$

Note que as inequações das restrições representam um conjunto de hiperplanos, de modo que todas as amostras (representadas por vetores com origem em O) satisfazem a equação de algum dos hiperplanos, formando então um conjunto de restrições para que a maximização da margem resulte em um plano de separação adequado. Note também que as amostras das bordas, onde os hiperplanos são iguais a 1 ou a -1 são considerados vetor de suporte.

Assim, temos a Equação (2.4) formalizando um problema de otimização, de objetivo quadrático e restrições lineares, solucionando o problema de classificação para dados linearmente separáveis.

$$\min\{\frac{1}{2}\|\mathbf{w}\|^2\} \quad \text{s.t.} \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \quad i = 1,\dots,N \quad (2.4)$$

2.1.5 SVM Formulação Dual

Podemos reescrever a formulação primal utilizando o operador lagrangiano. Essa nova formulação nos traz inúmeros benefícios. Permite expressar a formulação usando o produto interno das amostras permitindo o uso da função de *kernel* possibilitando modelar superfícies de decisão não lineares usando a formulação não linear.

A equação lagrangiana para otimização de função objetivo com múltiplas equações e múltiplas inequações de restrição é dada da seguinte forma: Tome o problema primal de maximizar a função $f(x)$ com múltiplas inequações $g_j(\mathbf{x}) = 0$ para $j = 1, \dots, J$ e múltiplas equações $h_k(\mathbf{x}) \geq 0$ para $k = 1, \dots, K$.

Colocando na forma de função Lagrangiana temos, onde α são multiplicadores de Lagrange das inequações e β são multiplicadores de Lagrange das equações e estão em forma de

vetores, sendo α_j e β_k componentes desses vetores.

$$L(\mathbf{x}, \alpha, \beta) = f(\mathbf{x}) + \sum_{j=1}^J \alpha_j g_j(\mathbf{x}) + \sum_{k=1}^K \beta_k h_k(\mathbf{x}) \quad (2.5)$$

Como desejamos minimizar a função objetivo e maximizar os multiplicadores de Lagrange, colocamos o sinal de menos nos multiplicadores (α), além disso, na formulação primal não há a necessidade de se representar a componente referente as equações, resultando em:

$$L(\mathbf{x}, \alpha) = f(\mathbf{x}) - \sum_{j=1}^J \alpha_j g_j(\mathbf{x}) \quad (2.6)$$

Aplicando a formulação do SVM primal temos em um primeiro passo a seguinte equação lagrangiana:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{j=1}^N \alpha_j (y_j (\mathbf{w}^T \mathbf{x}_j + b) - 1), \quad \text{onde } N \text{ é o tamanho da base de treinamento.} \quad (2.7)$$

Na literatura, alguns autores descrevem a função objetivo como $\frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2} \mathbf{w}^T \mathbf{w} = \frac{1}{2} \sum_{i=1}^{n_1} w_i^2$, onde n_1 é o tamanho do vetor \mathbf{w} e do vetor \mathbf{x} . Pode ser verificado observando na Figura 4 dado que os vetores \mathbf{x} e \mathbf{w} pertencem ao mesmo espaço.

Agora, basta calcular a derivada parcial da Equação (2.7) com relação a b e a \mathbf{w} .

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0 \quad \Rightarrow \quad - \sum_{j=1}^N \alpha_j y_j = 0 \quad \Rightarrow \quad \sum_{j=1}^N \alpha_j y_j = 0 \quad (2.8)$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} - \sum_{j=1}^N \alpha_j y_j \mathbf{x}_j = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{j=1}^N \alpha_j y_j \mathbf{x}_j \quad (2.9)$$

Aplicando a Equação (2.8) e (2.9) em (2.7) chegamos a seguinte equação (2.10).

$$L(\alpha) = -\frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^N \alpha_i \quad (2.10)$$

Assim, a formulação do problema de otimização primal pode ser reescrito da seguinte forma:

$$\max \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right\} \quad \text{s.t.} \quad \alpha_i \geq 0 \quad e \quad \sum_{i,j=1}^N \alpha_i y_i \quad i = 1, \dots, N \quad (2.11)$$

A função de predição para novos pontos, dada por $h(\mathbf{x})$, pode ser feita da seguinte maneira: $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$ com \mathbf{w} dado por $\mathbf{w} = \sum_{n=1}^N \alpha_n y_n \mathbf{x}_n$

$$h(\mathbf{x}) = \sum_{n=1}^N \alpha_n y_n \mathbf{x}_n^T \mathbf{x} + b \quad (2.12)$$

As condições de KKT (*Karush-Kuhn-Tucker*) para este tipo de problema de otimização deve satisfazer as seguintes condições, onde o vetor \mathbf{x}_n representa cada ponto dos dados.

$$\begin{aligned} \alpha_n &\geq 0 \\ y_n h(\mathbf{x}_n) - 1 &\geq 0 \\ \alpha_n \{y_n h(\mathbf{x}_n) - 1\} &= 0 \end{aligned} \quad (2.13)$$

2.1.6 SVM para Margens Suaves

A formulação primal para margens suaves adiciona o termo ξ para permitir a obtenção da região de decisão mesmo quando existem sobreposição de dados, além de tornar o método mais robusto à ruído (*outliers*).

Veja que o objetivo agora é minimizar $\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$ penalizando os pontos que estão do lado errado do hiperplano de separação.

$$\min \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right\} \quad \text{s.t.} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad i = 1, \dots, N \quad \xi \geq 0 \quad (2.14)$$

Abaixo segue a formulação Dual para Margens Suaves.

$$\max \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right\} \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C \quad e \quad \sum_{i=1}^N \alpha_i y_i = 0 \quad i = 1, \dots, N \quad (2.15)$$

2.1.7 SVM Kernel-Trick

Na formulação Dual do SVM podemos observar o produto interno $\mathbf{x}_i^T \mathbf{x}_j$, que pode ser substituído por $K(\mathbf{x}_i, \mathbf{x}_j)$, em que $K(\cdot, \cdot)$ é a função do *kernel*. Seu uso permite que algoritmos lineares solucionem problemas não lineares.

Abaixo estão listados as funções de *kernel* mais comuns:

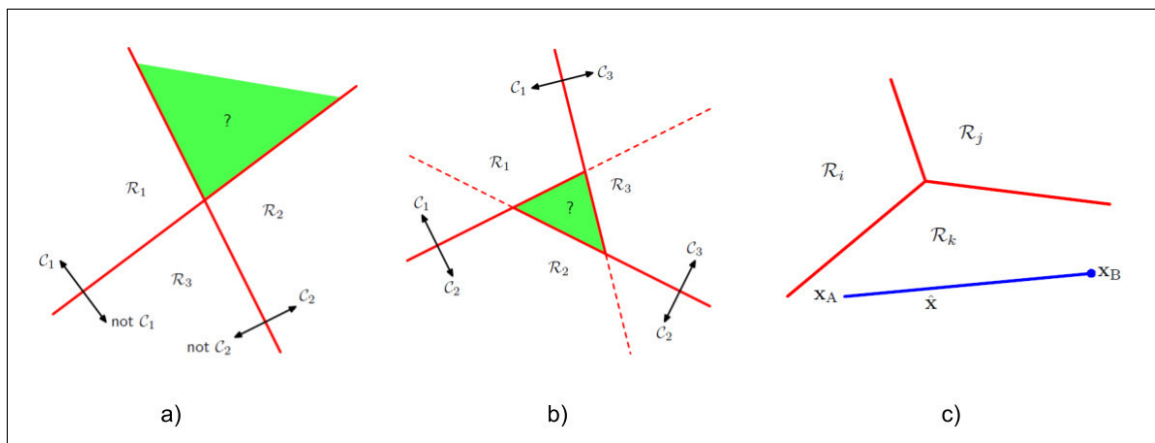
- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polinomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\alpha \mathbf{x}_i^T \mathbf{x}_j + c)^p$
- Gaussiano: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
- Exponencial: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|)$
- Sigmoidal: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(k \mathbf{x}_i^T \mathbf{x}_j + \delta)$

2.2 Classificação com Múltiplas Classes

Uma outra questão a ser abordada sobre classificadores diz respeito ao número de classes que se deseja classificar. Inicialmente o SVM foi proposto para tratar de problemas binários, mas pode ser estendido para problemas com número de classes maiores que 2, $K > 2$. O KNN, por verificar os vizinhos e votar pelo rótulo da maioria, se adapta mais facilmente em problemas com múltiplas classes.

Existem diferentes métodos para se modelar um classificador para múltiplas classes, existindo também algumas dificuldades a serem analisadas, ilustradas na Figura 6 (BISHOP, 2006) para o caso de um classificador com três classes. Por exemplo, podemos utilizar uma estratégia um-contra-todos (termo em inglês: *one-versus-all* ou *one-versus-rest*). Nessa estratégia $K - 1$ classificadores binários são treinados tendo em foco uma classe i em específico, enquanto que as outras classes são agrupadas como não sendo da classe i . O problema dessa abordagem, como podemos ver na figura 6a) é a existência de regiões ambíguas, como a região verde (R_1 e R_2 ao mesmo tempo).

Figura 6 – Múltiplas Classes



Fonte – Adaptado de (BISHOP, 2006)

Ao invés de usar a estratégia um-contra-todos, outra opção seria usar um-contra-um, que considera duas classes por vez dentre todos os pares de classes possíveis ignorando todas as outras e ao final a classe é estabelecida pelo voto majoritário. Essa estratégia gera $K(K-1)/2$ classificadores binários e também gera região ambígua como ilustrado na figura 6b).

Outra abordagem para evitar os problemas mencionados utiliza K classificadores e K funções lineares descrita pela equação (2.16), e a nova amostra é associada a classe C_k se

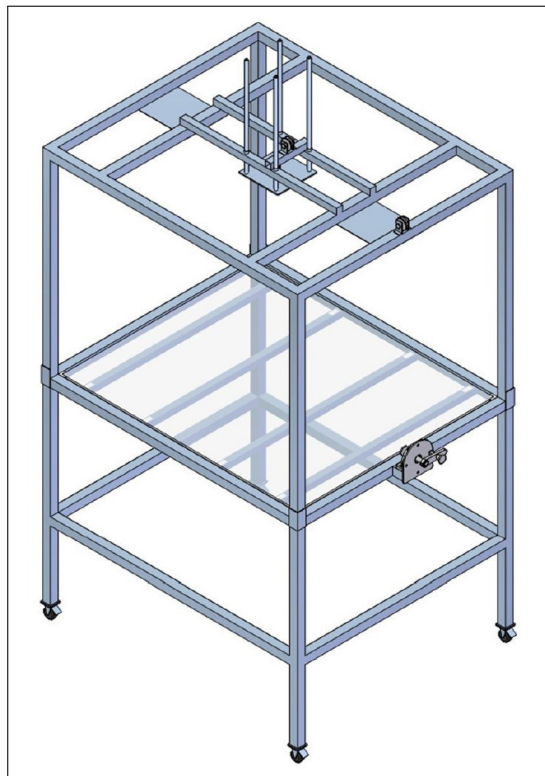
$h_k(\mathbf{x}) > h_j(\mathbf{x})$ para todos $j \neq k$. A região de decisão é mostrada na figura 6c), sendo cada região convexa (BISHOP, 2006).

$$h_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (2.16)$$

3 AQUISIÇÃO DE DADOS

As imagens deste trabalho foram obtidas por Santos Filho (SANTOS FILHO, 2013). Em sua dissertação de mestrado o autor descreve todo o processo industrial de tratamento do couro e também uma mesa para aquisição de imagens conforme pode ser visto na Figura 7. As dimensões da mesa são de 1,00m x 1,20m que, em geral atende as restrições de tamanho das peças trabalhadas pela indústria de curtição. Ao topo uma câmera é fixada para padronização das capturas.

Figura 7 – Mesa de Aquisição das Imagens



Fonte – (SANTOS FILHO, 2013)

Um exemplo de imagem capturada pelo mecanismo é demonstrada na Figura 8. A imagem é capturada depois de ter passado por processos curtição, estando no estado chamado de *Wet-Blue*, onde a peça de couro foi modificada quimicamente para obter resistência recebendo a coloração azulada (*blue*) devido aos produtos químicos aplicados e ficando com um aspecto úmido (*wet*).

A peça é colocada sobre um fundo preto para facilitar a extração da região de interesse. A imagem é então convertida para escala de cinza e um pré-processamento é feito para desconsiderar os pixels de fundo.

Figura 8 – Imagem Capturada



Fonte – (SANTOS FILHO, 2013)

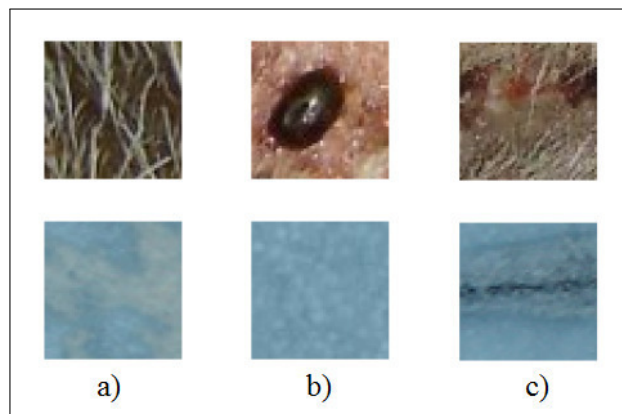
O aparato de captura de imagens nos traz a garantia de distância, necessário para manter a escala da imagem. Também é importante manter as configurações da câmera. No caso, as imagens foram capturadas com os mesmos padrões de fotografia, incluindo distância focal de 18mm, tempo de exposição de 1/50s e nível de sensibilidade ISO-200.

A rotação da imagem é outra variação que temos levar em consideração, principalmente para o método de extração de atributos conforme será visto no Capítulo 4. Para algumas aplicações a mudança na rotação é um efeito desejável. Para a aplicação deste trabalho, a classificação da qualidade de uma peça deve ser a mesma independente da rotação da fotografia. Para diminuirmos a influência da rotação, podemos trabalhar na padronização no momento da captura da imagem, ou podemos utilizar um método de extração invariante à rotação. Note que ter um método invariante à rotação nos dá mais garantias matemáticas para a classificação, enquanto que garantir que a foto foi retirada de maneira correta está sujeito a fatores externos, como o modo de operação da máquina de captura. Além disso, um mesmo tipo de falha em uma superfície pode ter diferentes rotações mesmo com as imagens possuindo o mesmo padrão de angulação.

Quanto a escolha das imagens para formar a base de dados, esta deve ser feita de maneira que as peças de couro representem todas as classes de qualidade possível para melhor generalização do problema. Ao todo, a base de dados foi composta por 350 imagens, divididas de modo que cada uma das 7 classes de qualidade seja representada por 50 imagens.

A origem das falhas na superfície do couro refletem na textura da imagem e podem ser as mais diversas, podendo ser arranhões ocorridos no campo ou mesmo causado por algum tipo de parasita, como carrapato, a Figura 9 apresenta algumas dessas falhas para o couro bovino, a imagem superior equivale a imagem da pele natural do animal e a inferior o couro após passar pelo processo químico, no estado *Wet-Blue*. (AMORIM, 2009)

Figura 9 – Exemplo de defeitos em couro bovino. Imagem superior origem do defeito. Imagem inferior o resultado no estado *Wet-Blue*. a) defeito sarna, b) defeito carrapato, c) defeito risco.



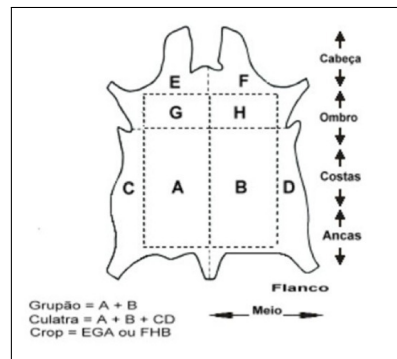
Fonte – Adaptado de (AMORIM, 2009)

Cada classe de qualidade está associada a presença de defeitos na peça de couro. Porém, na abordagem deste trabalho não estamos interessados em detectar cada tipo de avaria, isto tornaria o foco diferente, pois precisaríamos rotular cada tipo de falha. O que está sendo proposto, no entanto, é uma maneira para identificar as classes de qualidade em superfícies apenas com a informação de textura, abstraindo em parte os detalhes do significado de cada falha.

Para contextualizar o domínio do problema, apresentamos a seguir a Figura 10 e a Tabela 2. A Figura 10 apresenta as regiões específicas da peça de couro e ajuda a entender alguns termos mencionados pela Tabela 2 como os termos grupão (região central inferior) e flanco (regiões laterais).

A Tabela 2 tenta apenas agrupar visualmente as principais diferenças de cada classe

Figura 10 – Regiões do couro



Fonte – (SANTOS FILHO, 2013)

de qualidade apresentadas em Santos Filho. Apesar de serem razoavelmente bem definidas, na prática a identificação das características visuais tem se mostrado subjetiva por ser uma atividade que depende da interpretação humana (SANTOS FILHO, 2013). A tabela também apresenta alguns termos que estão abreviados por restrição de espaço, como 'Área de Aproveitamento' (Área de Apr.) e 'Proporcional ao Tamanho' (Prop.). Alguns termos técnicos como 'Flor' e 'Flanco' também são citados, sendo que flor é a área externa do couro, a mesma região da pelagem, que pode se deteriorar por fatores como arranhões ou ferimentos de parasitas como carrapato e flanco a região lateral do couro conforme especificado na figura 10.

Os valores das tabelas mais subjetivos foram divididos entre alta, média, baixa, qualquer, nenhuma e proporcional. Dependendo da característica o uso do nível 'qualquer' é mais apropriado do que o nível 'nenhuma'. Algumas características apresentam o valor numérico acompanhadas de uma descrição já outras apresentam um valor de porcentagem.

Tabela 2 – Parâmetros para classificação manual de couro

Características	Classes						
	1	2	3	4	5	6	7
Área de Apr. Mestiços	-	>90%	85 a 90%	70 a 85%	55 a 70%	40 a 55%	30 a 40%
Área de Apr. Cabra	-	>90%	75 a 85%	60 a 75%	45 a 60%	30 a 45%	25 a 35%
Exigência de formato de pele	-	Alta	Alta	Média	Baixa	Nenhuma	Nenhuma
Nível de Deterioração da flor	-	Nenhum	Baixa	Média	Qualquer	Qualquer	Qualquer
Nível Arranhões e Cicatrizes	-	Nenhum	Baixa	Baixa	Qualquer	Qualquer	Qualquer
Exigência de pele encorpada	-	Alta	Prop.	Prop.	Baixa	Nenhuma	Nenhuma
Tolerância de cortes de faca	-	Baixa	Baixa	Média	Qualquer	Qualquer	Qualquer
Nº de furos pequenos	-	1 no flanco	2 no grupão	3 no grupão	3 no grupão	Qualquer	Qualquer
Nº Max. de Furos grandes	-	0	1 (<10cm)	1(<10cm)	1(<10cm)	3(>10cm)	Qualquer
Área de defeito	-	0%	<5%	<10%	<20%	Qualquer	Qualquer

Fonte – Baseado em (SANTOS FILHO, 2013)

Assim, das três etapas processo de classificação, este capítulo representou os principais elementos da etapa de Aquisição de Dados, levando em consideração os aspectos como o contexto do problema e captura de imagens. Aspectos como padronização e os problemas envolvidos na aquisição das imagens também foram discutidos. O próximo capítulo irá discutir a etapa seguinte, focando nos métodos para extração de atributos das imagens, no caso os método de padrão binário local (LBP).

4 EXTRAÇÃO DE CARACTERÍSTICAS USANDO LBP

Após a aquisição dos dados, a segunda etapa do processo de classificação consiste na extração de características. No caso deste trabalho, estamos interessados em categorizar diferentes tipos de qualidade de peças de couro, mais precisamente 7 tipos de classes. É natural pensar que um método descritor de texturas seja a opção mais adequada visto que cada classe está associada a danos na superfície do couro, o que irá refletir na variação da textura da superfície. No caso, o extrator escolhido para análise foi o Padrão Binário Local.

Este capítulo iniciará fazendo uma breve introdução sobre o reconhecimento de texturas e as seções seguintes serão direcionadas a explicar o LBP e suas variações mais tradicionais, a saber: LBP Uniforme, LBP Invariante a Rotação e LBP Uniforme e Invariante à Rotação.

4.1 Textura em Processamento de Imagens

A textura está frequentemente ligada a percepção humana, não existindo assim uma definição matemática única. A textura pode ser definida como se queira, e de acordo com o problema (NIXON; AGUADO, 2012).

Petrou e Sevilla afirmam que textura é uma variação de dados em uma escala menor que a escala de interesse, assim, padrões que em um momento podem ser descritos como objetos, em outra escala poderão ser resumidos em forma de textura (PETROU; SEVILLA, 2006).

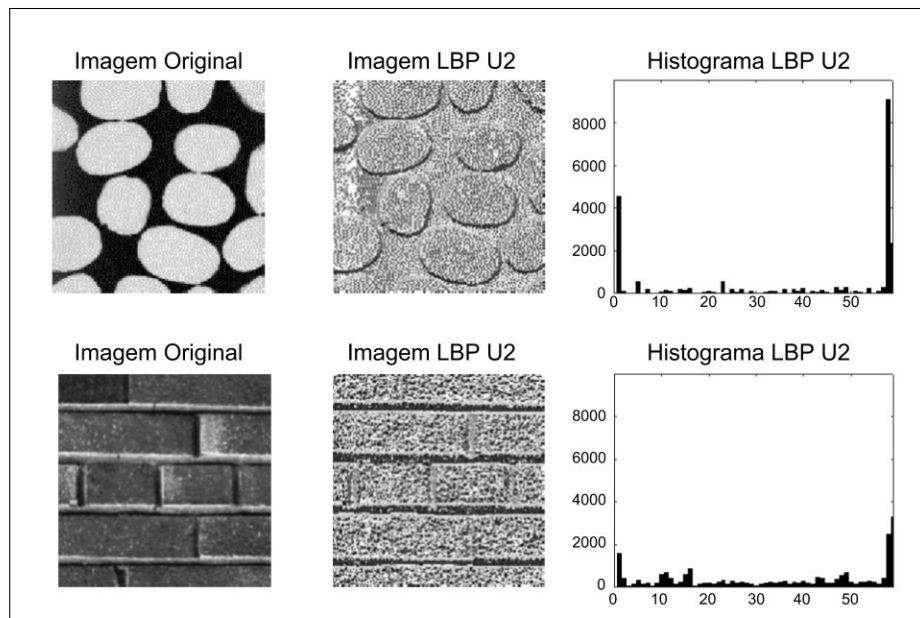
Classificar imagens requer fazer com que o computador seja capaz de interpretar elementos úteis para a discriminação de imagens que compartilham características em comum. Neste sentido, a extração de atributos se torna uma etapa importante para que a classificação seja bem sucedida. Diversos métodos em visão computacional focam na identificação de algum tipo de características da imagens. Reconhecer faces, detectar de sorrisos, extrair olhos vermelhos, identificar objetos e extrair de plano de fundo exemplificam tarefas com diferentes propriedades que devem ser identificadas na imagem.

O reconhecimento de texturas tem destaque especial devido a sua característica natural de possibilitar a diferenciação de regiões em imagens. Assim, extrair informação de textura se torna uma tarefa útil para solução de diversos problemas envolvendo análise de imagens, não se limitando apenas à identificação de texturas em si.

4.2 Padrão Binário Local (LBP)

O Padrão Binário Local consiste em um descritor de texturas que realiza uma operação matemática sobre cada pixel, gerando uma nova imagem de padrões binários locais. A frequência (histograma) dos padrões na imagem resultante da operação LBP é normalmente usada como vetor de características para a descrição de texturas. Para se ter uma ideia do resultado da extração usando o método LBP a Figura 11 exemplifica dois tipos de texturas com suas respectivas extrações da imagem LBP e do histograma LBP.

Figura 11 – Exemplo de texturas, com a extração de suas respectivas imagens e histograma LBP



Fonte – O Autor, Texturas utilizadas: (BRODATZ, 1999)

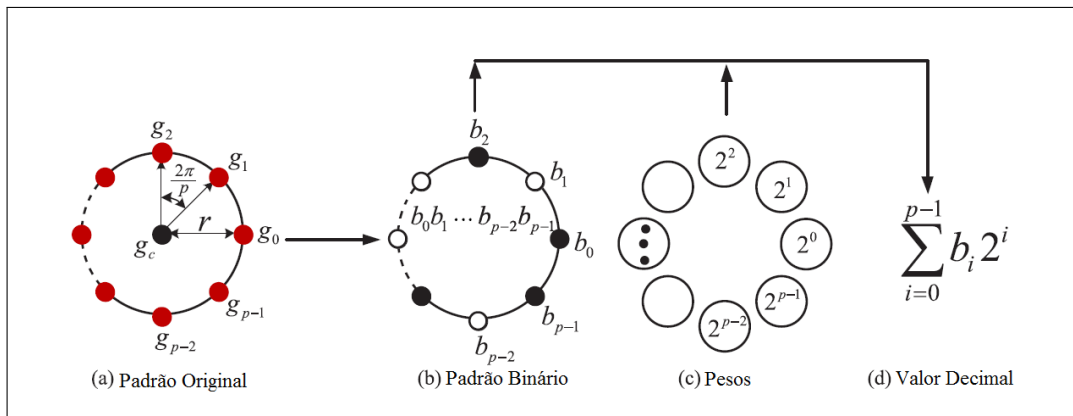
O operador LBP foi inicialmente proposto por Ojala (OJALA *et al.*, 1994) como uma modificação do método de análise de textura de Wang e He (WANG; HE, 1990a), (WANG; HE, 1990b), usando apenas dois níveis de comparação com pixel central ao invés de três e ambos estavam limitados como operador de tamanho 3x3. Em 2002 (OJALA *et al.*, 2002) o operador LBP foi generalizado permitindo qualquer quantidade de pontos de vizinhança e qualquer valor de raio, ou seja, permitindo a variação dos parâmetros de extração. Outra contribuição de seu trabalho de 2002 foi ter constatado que existem padrões binários que descrevem propriedades fundamentais de textura local, os chamados padrões uniformes. Além disso, Ojala *et al.* também propôs a versão invariante à rotação, e a versão uniforme e invariante à rotação.

Atualmente o operador LBP tem sido extensamente estudado devido sua capacidade

de descrever texturas e atualmente possui diversas propostas de extensão e variação da técnica, que se adaptam a diferentes tipos de aplicações, podendo fazer parte da solução de problemas em reconhecimento de faces, detecção de objetos e extração de característica em vídeos.

A extração do LBP básico consiste em analisar os píxels da vizinha de um pixel central. Os píxels da vizinhança que tiverem a intensidade maior ou igual que a do pixel central são associados ao valor 1, e os píxels de intensidade menor são associados ao valor 0. Cada valor binário é ponderado pela posição de modo a formar um valor binário, que será convertido em um valor inteiro ≥ 0 . A figura 12 apresenta geometricamente a extração do LBP para melhor entendimento da formalização matemática.

Figura 12 – Representação geométrica do formalismo da Extração LBP



Fonte – Traduzido de (LIU *et al.*, 2017)

A formalização matemática segue da seguinte maneira. Os P píxels da vizinhança são identificados por $g_p(x_p, y_p)$ em uma região da circunferência de raio R , com $p = \{0, \dots, P-1\}$. O píxel central é identificado como $g_c = (0, 0)$. A formulação do posicionamento de cada píxel da vizinhança é dada por (x_p, y_p) onde, $x_p = -R \sin(\frac{2\pi p}{P})$, e $y_p = R \cos(\frac{2\pi p}{P})$. Desta maneira, o LBP pode ser definido como segue

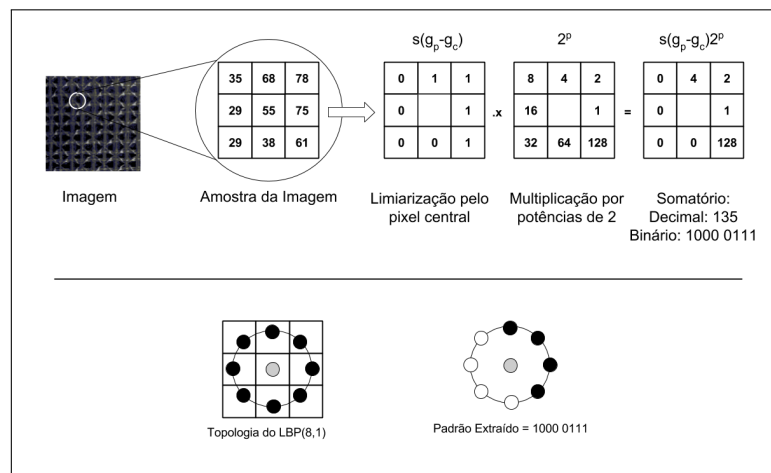
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \text{ onde } s(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases} \quad (4.1)$$

Como podemos ver na Figura 12a), o LBP é extraído a partir de uma topologia em que os píxels da vizinhança estão localizados sobre uma circunferência de raio R , e igualmente espaçados. A Figura 12b), representa a etapa após a comparação dos píxels da vizinhança com o píxel central. Como foi formalizado, aqueles píxels g_p maiores ou iguais a g_c são atribuídos

o valor 1 (pontos pretos), 0 (pontos brancos) caso contrário. Na Figura 12c), podemos ver a associação de cada posição dos bits com os pesos na potências de 2 sobre a topologia circular, a fim de gerar um valor na base 10, representado matematicamente pela Figura 12d).

Para exemplificar esse processo, a Figura 13 descreve a extração o LBP de um pixel usando 8 pontos vizinhos e 1 pixel de raio (denotado por $LBP_{8,1}$). Na parte superior temos uma sequencia de cinco imagens. A primeira corresponde a imagem original da Textura em escala de cinza. A segunda imagem apresenta uma amplificação na região do pixel, mostrando o pixel central e os pixels da vizinhança. A terceira imagem ainda na Figura 13 apresenta o resultado da comparação da intensidade dos pixels da vizinhança com o a intensidade do pixel central. Note que o valor binário associado aos pixels da vizinhança com valores maiores ou igual ao do pixel central receberam valor 1, enquanto os com intensidade menor receberam valor 0. Veja também que a topologia para o $LBP_{8,1}$ gerou o formato de um quadrado com 3x3 pixels.

Figura 13 – Ilustração da Extração LBP



Fonte – O Autor

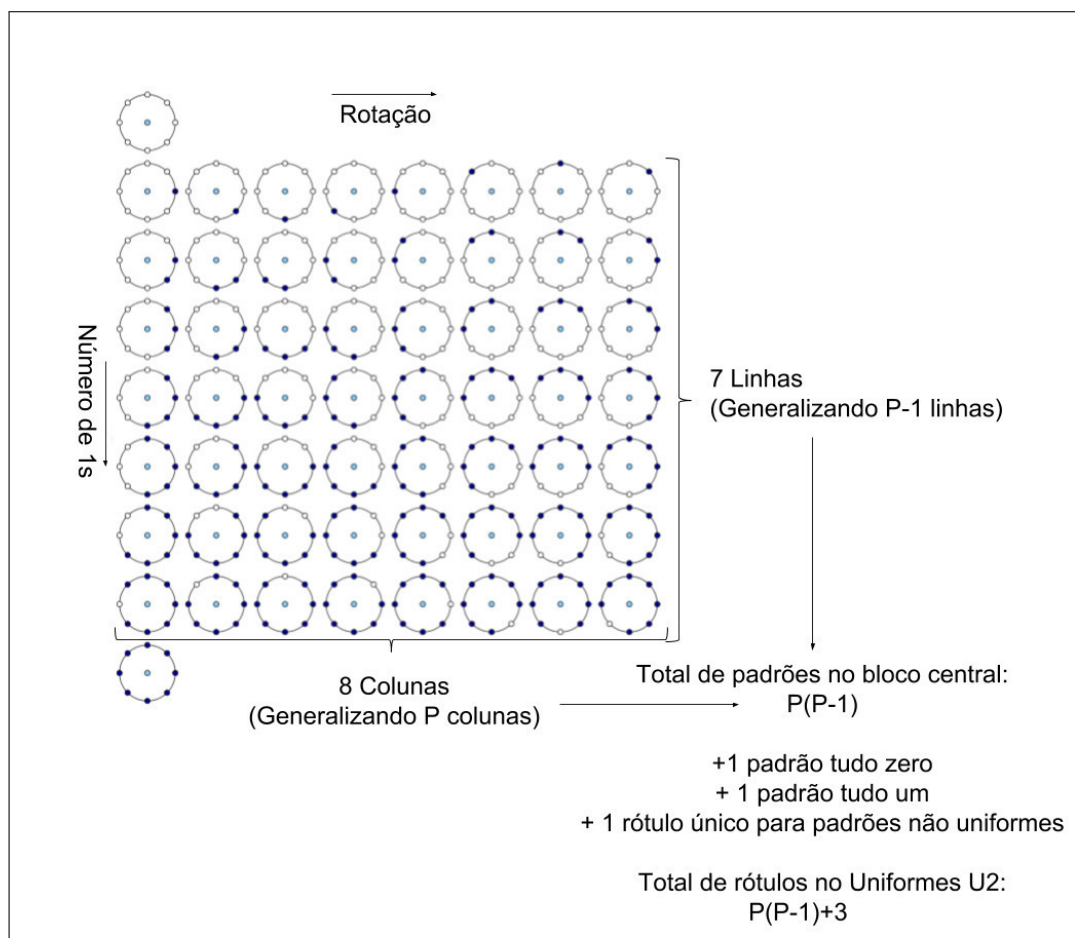
A Figura 13 apresenta ainda o padrão extraído, 10000111_2 , ou 135_{10} , que coincidentemente exemplifica um padrão uniforme, possuindo zeros e uns contíguos na sequência binária circular. Esse tipo de padrão será visto mais em detalhes na Seção 4.4. A quantidade de padrões gerados pelo LBP básico é de 2^P , exponencial no número de pontos. Porém o LBP Uniforme reduz essa quantidade para a ordem de P^2 e o LBP Uniforme e Invariante a Rotação reduz a quantidade para a ordem de P .

As próximas seções irão abordar o padrão binário local uniforme ($LBP_{P,R}^{u2}$), o padrão binário local invariante à rotação ($LBP_{P,R}^i$) e o padrão binário local invariante à rotação e uniforme ($LBP_{P,R}^{riu2}$).

4.3 Método LBP Uniforme

O termo uniforme está relacionado aos padrões LBP com a propriedade especial de possuir blocos de 0 ou de 1 contíguos formando os chamados padrões binários uniformes. Descrevendo de maneira mais formal, esses padrões, ou sequencia de bits, possuem poucas transições de 0 para 1 ou 1 para 0, normalmente no máximo duas transições, levando em consideração a sequencia circular de bits e sendo denotado por LBP^{u2} . Em 2002 Ojala *et al.* constatou experimentalmente que cerca 90% de todos os padrões 3×3 ($LBP_{8,1}$) são formados pelos padrões uniformes. O uso do LBP Uniforme diminui consideravelmente o número de padrões, se com o LBP básico, a quantidade estava na ordem de 2^P , com o LBP Uniforme passa para a ordem de P^2 . Operador LBP Uniforme detecta esses padrões uniformes e dá novos rótulos para cada padrão.

Figura 14 – Padrões uniformes para 8 pontos.



Fonte – Adaptado de (AHONEN *et al.*, 2009)

A Figura 14 apresenta os padrões uniformes para 8 pontos de vizinhança, existindo

no máximo duas transições de 0 para 1 ou 1 para 0. Na ilustração, cada linha i (começando da linha zero) apresenta padrões contendo i pontos com valor 1 e suas devidas rotações. Podemos verificar a quantidade de padrões gerados pelo $LBP_{P,R}^{u,2}$ através de uma generalização da Figura 14, note que as 7 linhas centrais possuem 8 rotações, gerando 56 padrões, generalizando, temos $P(P-1)$ padrões. Além disso, temos a linha 0 representando o padrão "tudo zero", e a linha 8 representando o padrão "tudo um", e qualquer outro padrão diferente é rotulado de maneira única. Totalizando 59, ou de maneira geral $P(P-1) + 3$ padrões.

Figura 15 – Determinando o número de transições de 0 para 1 ou 1 para zero. Exemplificando a aplicação da Equação (4.2) para uma sequência de 8 bits. A sequência foi espaçada a cada 4 bits para facilitar a visualização.

Exemplo 1

Padrão Binário: $0_7 \quad 1_6 \quad 1_5 \quad 0_4 \quad 0_3 \quad 1_2 \quad 1_1 \quad 0_0$ Índices

Padrão Binário:

0_7	1_6	1_5	0_4	0_3	1_2	1_1	0_0
-1_6	-1_5	-0_4	-0_3	-1_2	-1_1	-0_0	-0_7
-1	0	-1	0	-1	0	1	0

Padrão binário com Deslocamento Circular para esquerda:

$|-1| + |0| + |-1| + |0| + |-1| + |0| + |1| + |0| = 4$ Número de Transições 0 para 1

Exemplo 2

Padrão Binário: $0_7 \quad 0_6 \quad 0_5 \quad 1_4 \quad 1_3 \quad 1_2 \quad 1_1 \quad 1_0$

0_7	0_6	0_5	1_4	1_3	1_2	1_1	1_0
-0_6	-0_5	-1_4	-1_3	-1_2	-1_1	-1_0	-0_7
0	0	-1	0	0	0	0	1

$|0| + |0| + |-1| + |0| + |0| + |0| + |0| + |1| = 2$

Fonte – O autor.

A uniformidade de um padrão binário é medida pelo operador U , como mostrado na Equação (4.2), que calcula a quantidade transição de 0 para 1 ou de 1 para 0 de uma sequência binária circular. Por exemplo, um padrão 00000000_2 e o 11111111_2 não possui transição,

portanto, o valor U é 0. Já o padrão 00011110_2 possui valor U igual a 2 por possuir duas transições. A extração dos padrões uniformes $LBP_{P,R}^{u2}$ considera no máximo duas transições. A contagem de transições realizadas pela Equação (4.2) acontece subtraindo a sequencia original dela mesma; porém, com um deslocamento circular para a esquerda, como mostrado na Figura 15. Em outras palavras, um bit na posição P é subtraído pelo bit na posição $P - 1$ e então contabilizado o seu módulo. Podemos também ver a primeira componente da Equação (4.2) de U como uma exceção, onde o bit da posição zero não subtrai o anterior e sim o bit $P - 1$, o mais significativo da sequencia.

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (4.2)$$

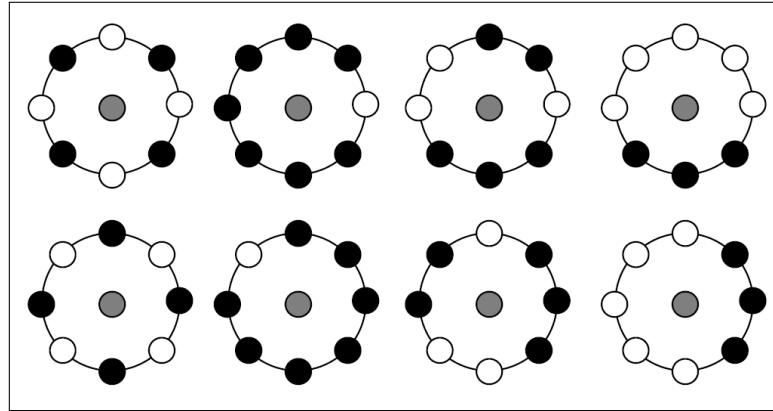
Em resumo, o operador $LBP_{P,R}^{u2}$ identifica os padrões com no máximo duas transições (01 ou 10) através do operador U e atribui a esses padrões novos rótulos. Aos padrões não uniformes é dado um rótulo único. Assim, o $LBP_{P,R}^{u2}$ consegue diminuir a quantidade de padrões de 2^P no LBP básico para $P(P - 1) + 3$. Além de reduzir a dimensionalidade do LBP, a uniformização também dá mais robustez contra ruído, pois quando se considera apenas os padrões LBP contendo apenas blocos de zeros e uns contíguos, sendo essas, propriedades importante para o presente trabalho e será citada nos experimentos.

4.4 Método LBP Invariante à Rotação

Invariância à rotação, consiste em diminuir os efeitos causados pela rotação da imagem na extração de características ou, de uma maneira ideal, em eliminar esse efeito conseguindo obter o mesmo padrão de característica mesmo quando as imagens estão com rotações diferentes.

A invariância a Rotação do $LBP_{P,R}^{ri}$ podemos dizer que vai mais além de uma simples rotação da imagem, nesse método cada pixel é considerado um ponto de rotação. Alguns padrões binários quando rotacionados podem equivaler a padrões semelhantes. Tome por exemplo o padrão 00010000_2 e o padrão 00000001_2 , através de 4 deslocamento circular para a direita no primeiro padrão é possível se obter o segundo. A função $ROR(x, i)$ faz i deslocamento dos bits no padrão binário x e é usada pelo $LBP_{P,R}^{ri}$ para fazer com que a maior quantidade de zeros estejam nas posições mais significativas. Em outras palavras, a função é usada para minimizar o valor do padrão binário. Padrões equivalentes recebem o mesmo rótulo e assim pode-se diminuir a quantidade de padrões possíveis e tornar o método invariante a rotação.

Figura 16 – Exemplo de padrões invariantes à rotação. A linha superior é equivalente a linha inferior.



Fonte – Adaptado de (MÄENPÄÄ, 2003)

Em uma linguagem mais informal podemos representar a função $ROR(x, i)$ como o deslocamento de bits circular, mas para se ter uma definição mais precisa, iremos definir o padrão binário e, em seguida, a função $ROR(x, i)$ e o $LBP_{P,R}^i$ seguindo a definição usada por Mäenpää (2003) em sua tese de doutorado (MÄENPÄÄ, 2003).

Um valor inteiro, pode ser definido em termos de uma sequência binária a_k , $k \in \{0, \dots, P-1\}$ e é apresentada na Equação (4.3).

$$x = \sum_{k=0}^{P-1} 2^k a_k, \text{ com } a_k \in \{0, 1\} \quad (4.3)$$

Já a função $ROR(x, i)$ está definida em 4.4.

$$ROR(x, i) = \begin{cases} \sum_{k=i}^{P-1} 2^{k-i} a_k + \sum_{k=0}^{i-1} 2^{P-i+k} a_k, & i > 0 \\ x, & i = 0 \\ ROR(x, P+i) & i < 0 \end{cases} \quad (4.4)$$

E, por fim, a definição do $LBP_{P,R}^i$ segue em 4.5.

$$LBP_{P,R}^i = \min\{ROR(LBP_{P,R}, i) | i = 0, 1, \dots, P-1\} \quad (4.5)$$

Sobre a quantidade de padrões gerados pelo método invariante à rotação, para uma quantidade de pontos da vizinhança $P = 8$ a quantidade extraída é de 36 padrões diferentes. Segundo Mäenpää (2013), determinar a quantidade de padrões a partir do número P de pontos da vizinhança não é uma tarefa trivial e o algoritmo até então desenvolvido para realizar o cálculo leva tempo $O(P^2)$.

A operação se torna mais relevante quando aplicada em conjunto com a extração de padrões LBP uniformes. (OJALA *et al.*, 2002). A Equação (4.5) descreve o LBP invariante a rotação e na seção 4.5, o método LBP Uniforme e Invariante a rotação.

4.5 Método LBP Uniforme e Invariante à Rotação

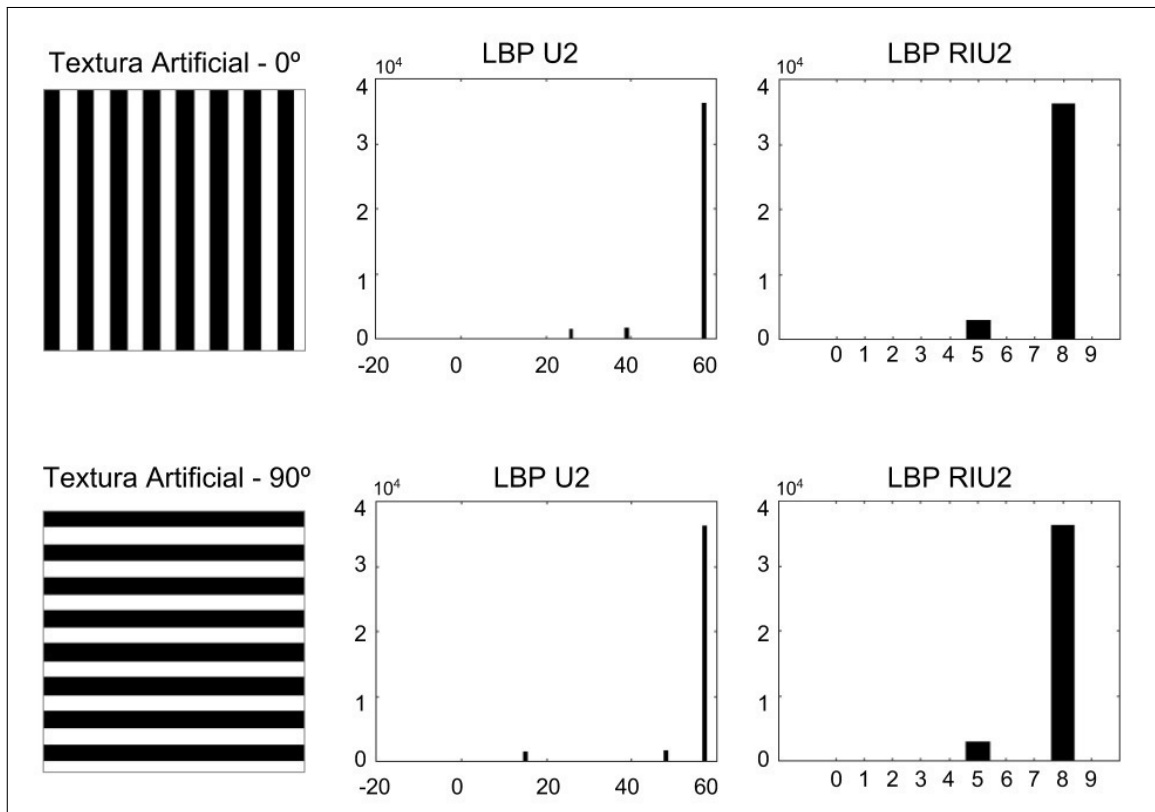
Se aplicarmos a invariância a rotação sobre os padrões uniformes, teremos o $LBP_{P,R}^{riu2}$. Podemos representar esses padrões através da primeira coluna da figura 14, formando um total de $P + 1$ padrões RIU2 (uniformes e invariante a rotação) e os demais padrões são rotulados com rótulo único, totalizando $P + 2$ rótulos. O uso do $LBP_{P,R}^{riu2}$ diminui consideravelmente a quantidade de padrões, conseqüentemente reduzindo a dimensionalidade além de dar maior robustez contra ruído ao método. A formulação matemática do $LBP_{P,R}^{riu2}$, consiste basicamente em contar quantidade de 1's em um padrão binário uniforme e classificar qualquer outro padrão com um rótulo único (P+1). A formalização matemática segue na Equação (4.6), onde U foi definida na Equação (4.2).

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & , \text{ Se } U(LBP_{P,R}) \leq 2 \\ P + 1 & , \text{ Caso Contrário} \end{cases} \quad (4.6)$$

Podemos comparar o método $LBP_{P,R}^{riu2}$ com $LBP_{P,R}^{u2}$, quanto a invariância à rotação. A Figura 17 apresenta os resultados da aplicação de cada operador sobre uma textura artificial em duas rotações. A primeira coluna apresenta a textura em dois ângulos diferentes, a segunda coluna mostra o histograma da aplicação do operador LBP Uniforme, e a terceira coluna que mostra o histograma do $LBP_{P,R}^{riu2}$. Note que o $LBP_{P,R}^{riu2}$ se manteve constante demonstrando a invariância dos padrões com a rotação da imagens, embora a demonstração seja feita com uma textura artificial e com ângulos de rotações grandes saltando de 0 para 90 grau, é importante ressaltar que menores ângulos de rotação exigirão maior número de pontos da vizinhança LBP, ou seja, a resolução angular, dada por $360^\circ/P$, pode melhorada com maior número P de pontos. Porém, Mäenpää (2003) lembra que não se pode aumentar P de qualquer maneira, pois a escolha de muitos pontos para valores de raio pequeno pode resultar em informação redundante, como por exemplo, em um raio de 1 pixel, usar um P maior de 8 não adicionaria informação a descrição da textura.

Para entendermos a extração do LBP resultante da Figura 17 iremos analisar em

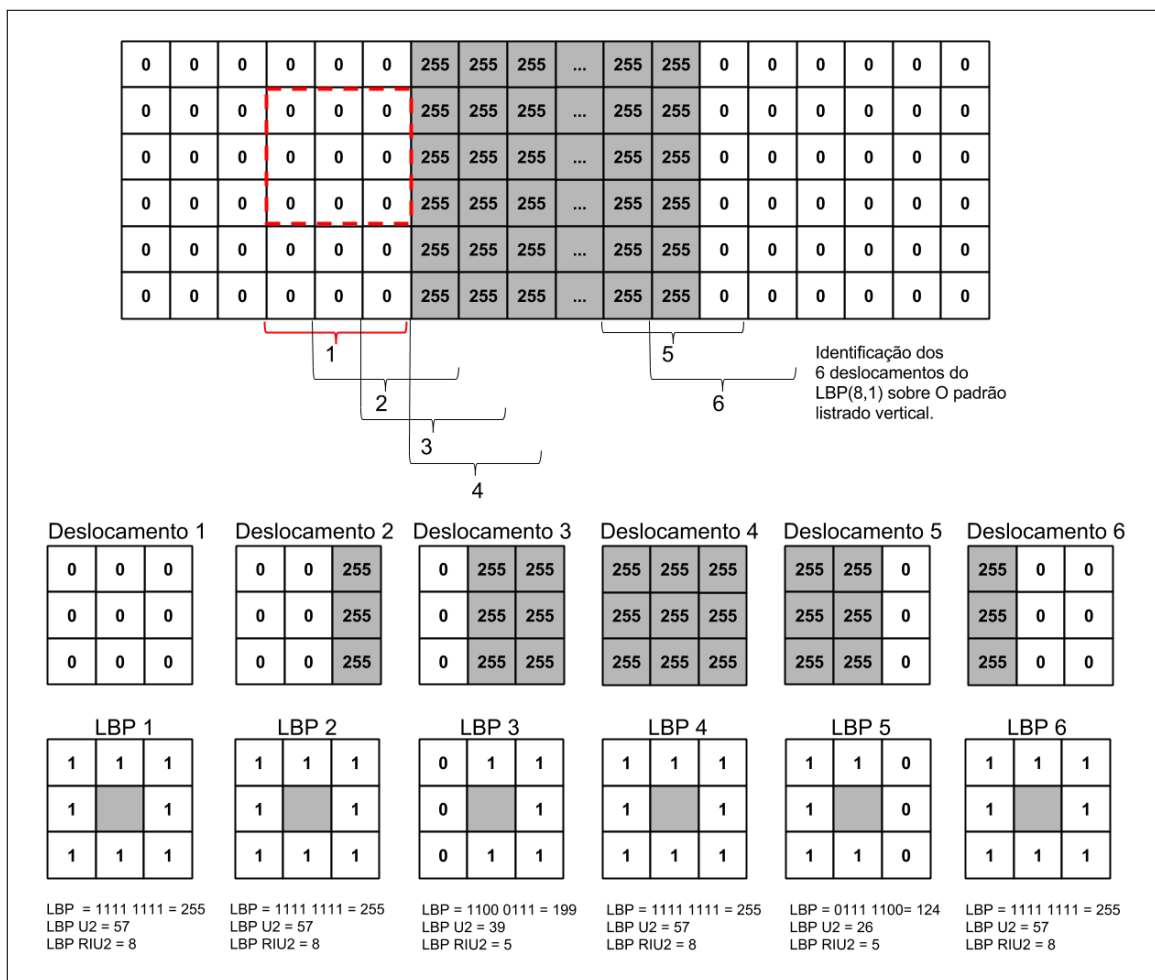
Figura 17 – Comparação entre LBP U2 e LBP RIU2 (Textura Artificial 200x200)



Fonte – O Autor

detalhes o deslocamento do operador $LBP_{8,1}$ sobre a imagem com de listras vertical. A figura 18 apresenta 6 deslocamentos que representam todos os padrões LBP básicos extraídos e em seguida sua codificação em LBP^{u2} e LBP^{riu2} . Observe que quanto o pixel central é zero, qual outro pixel é maior ou igual ao pixel central tornando os bits da vizinhança iguais a 1 no deslocamento 1, 2 e 6. Em uma análise superficial, poderia se pensar que o deslocamento 2 e 6, por apresentarem deslocamentos distintos também apresentariam valores LBP distinto para os casos em que não são invariantes a rotação, mas por se tratar de um pixel central com nível de cinza de valor zero, acaba sendo uma extração especial.

Figura 18 – Entendendo os padrões da textura artificial da Figura 17. Cada deslocamento representa os tipos de LBP(8,1) possíveis na textura de listras verticais. Note que quando o pixel central é zero, todos os pixels vizinhos são maiores ou iguais a ele.



Fonte – O Autor

5 CLASSIFICAÇÃO DA QUALIDADE DE COURO CAPRINO UTILIZANDO LBP

Uma aplicação para auxiliar na identificação da qualidade de couro caprino teria relevante papel para a indústria visto que a atividade de classificação determina o valor de cada peça. Por isso fazer uma classificação ágil, padronizada e objetiva (SANTOS FILHO, 2013) ajudaria a desenvolver a atividade.

O trabalho de Santos Filho (2013) apresenta uma primeira abordagem para a aplicação de classificadores de couro caprino, fazendo uso de GLCM (Matriz de co-ocorrências de níveis de cinza) e outros métodos para extração de características. Porém, a classificação foi feita simplificando de sete classes para apenas duas classes e, além disso, um estudo com LBP ainda não foi explorado. No intuito de preencher essa lacuna, este trabalho pretende avaliar o desempenho do LBP na classificação de diferentes qualidade de couro.

Foram apresentados nos Capítulos 2, 3 e 4 a fundamentação teórica para a classificação da qualidade das peças de couro caprino. Pretende-se, então, fazer uso dos classificadores KNN e SVM como métodos de aprendizagem de máquina e três métodos $LBP_{P,R}$ para extração de atributos das imagens. Pretende-se fazer uma busca pelos melhores parâmetros de extração (P,R) para se obter a melhor acurácia possível. Pretende-se, ainda, combinar os três $LBP_{P,R}$ mais simples em uma abordagem multi-resolução para verificar se ocorre melhorias na extração de características.

Na etapa de experimentos, inicialmente foi feito uma série de testes para verificar o desempenho dos operadores LBP quanto a capacidade de discriminação entre as classe de qualidade de couro caprino. Como cada operador possui parâmetros (P,R) para extração de atributos, os experimentos visaram a seleção dos melhores parâmetros a fim de se obter a maior acurácia. A classificação foi feita usando os métodos KNN e SVM em três abordagens de extração. Na primeira abordagem os operadores $LBP_{P,R}^{u2}$, $LBP_{P,R}^{ri}$ e o $LBP_{P,R}^{riu2}$ foram comparados entre si quanto ao desempenho da classificação, na segunda foi feita a combinação entre dois operadores, num processo de multi-resolução, onde são modificados (P,R) de maneira independente, e na terceira a multi-resolução foi feita combinados três operadores.

Na realização dos experimentos cada um dos operadores foi aplicado por vez sobre cada uma das imagens, gerando uma base de dados para cada alteração dos parâmetros do extrator com as características extraídas de cada imagem. Para a classificação, cada base foi dividida entre treinamento e teste e, então, aplicado o classificador KNN e o SVM com *kernel* polinomial usando a estratégia um-contra-um (*one-vs-one*) com código de correção de erro de saída para

múltiplas classes (*Error-correcting Output Codes*). Ao final foi feita uma tabela comparando os melhores resultados para os classificadores KNN e SVM para cada operador, como pode ser visto nas Tabelas 6, 9 e 12. Como se trata um problema com múltiplas classes, apenas média da taxa de acerto das classes é mostrada nas tabelas para a análise de desempenho. O operador LBP mais adequado a aplicação é então escolhido e sua matriz de confusão é mostrada para uma análise mais detalhada.

Quanto aos dados, a base de imagens utilizada foi composta por 350 imagens de couro caprino, sendo cada uma das 7 classes com 50 imagens. Após a aplicação de um operador LBP cada imagem passou a ser representada por um vetor de características, dado pelo histograma dos valores LBP. Então, as 350 imagens foram divididas entre $30 \times 7 = 210$ para treino e $20 \times 7 = 140$ para testes. Uma discussão mais detalhada de cada abordagem é feita nas seções seguintes.

5.1 Representação do Vetor de características

Os extratores LBP utilizados neste trabalho retornam o histograma da ocorrência dos padrões binários numa imagem. O histograma pode ser visto como um vetor em que cada componente do vetor representa quantidade de ocorrências de um determinado padrão binário do LBP. Como foi visto durante todo o capítulo 4, a quantidade de padrões, representado por valores na base 10, depende do número P de pontos da vizinhança e do tipo do LBP. Assim, do mesmo modo o tamanho do vetor varia com o valor de P , independentemente do tamanho da imagem ou do valor R do raio utilizado. Podemos representar um vetor \mathbf{v} de características LBP de acordo com a Equação (5.1) em que m representa a quantidade de padrões binários possíveis de extrair com o LBP em questão.

$$\mathbf{v} = \langle v_1, v_2, v_3, \dots, v_m \rangle^T \quad (5.1)$$

Para se ter uma ideia da dimensionalidade de cada extrator, apresentamos as Tabelas 3, 4, 5 com o tamanho dos vetores para cada tipo de LBP. A Tabela 3 apresenta todos os tamanhos dos vetores para os operadores sem combinação, enquanto que as tabelas 4 e 5 devido as várias possibilidades de combinação dos parâmetros (P, R) apresentam apenas os tamanhos mínimos e máximos possíveis para cada combinação de extrator.

Na Tabela 3 podemos ver que o $LBP_{P,R}^{ri}$ possui o maior número de parâmetros quando o número de pontos é igual a 16. Já o $LBP_{P,R}^{u2}$ apresenta uma quantidade intermediária de atributos. O $LBP_{P,R}^{riu2}$ apresenta menor quantidade de atributos. Vale ressaltar que o $LBP_{P,R}^{ri}$ é mais

Tabela 3 – Tamanho do vetor LBP sem combinação

	P=4	P=8	P=16
1.LBP_U2	15	59	243
2.LBP_RI	6	36	4116
3.LBP_RIU2	6	10	18

Fonte – O autor

susceptível a ruído, visto que não possui o critério de escolher apenas os padrões uniformes. Outra observação importante, dessa vez referente ao operador $LBP_{P,R}^{riu2}$, é que além de apresentar o menor número de atributos, ele ainda é invariante a rotação e mais imune a ruídos, tornando-se a melhor escolha para a aplicação.

Combinar operadores com parâmetros diferentes nos traz, ou pelo menos deveria trazer, mais informação sobre textura. A combinação de operadores foi feita concatenando-se os histogramas de cada operador simples, deste modo, tamanho do vetor é a soma dos vetores de cada operador simples. As Tabelas 4 e 5 apresentam a quantidade mínima e máxima de atributos que se pode obter combinando diferentes tipos de LBP.

Tabela 4 – Tamanho do vetor da combinação de 2 LBP

	Tam.Min P1=4,P2=4	Tam.Max P1=16,P2=16
1. U2_U2	30	486
2. RI_RI	12	8232
3. RIU2_RIU2	12	36
4. U2_RI	21	4359
5. U2_RIU2	21	261
6. RI_RIU2	12	4134

Fonte – O autor

A Tabela 4 apresenta a quantidade de características obtidas com a combinação de 2 operadores simples. Veja que a combinação de operadores com maior dimensionalidade é a $LBP_{P1,R1}^{ri} + LBP_{P2,R2}^{ri}$, representado na linha 2 da tabela e denodada por RI_RI, pois apresenta 8232 dimensões quando configurada com os parâmetros P1=16 e P2=16. Quanto à combinação $LBP_{P1,R1}^{riu2} + LBP_{P2,R2}^{riu2}$ representado pela linha 3, RIU2_RIU2, extrai no máximo 36 atributos, sendo a menor quantidade de atributos extraídos quando comparados com os outros métodos na configuração $P1 = 16, P2 = 16$.

Já na Tabela 5, podemos ver um dos principais motivos para não se usar o operador $LBP_{P1,R1}^{ri} + LBP_{P2,R2}^{ri} + LBP_{P3,R}^{ri}$, representado na tabela pela linha 2, RI_RI_RI. Veja que este

possui a maior dimensionalidade para $P1=16$, $P2=16$ e $P3=16$, tendo 12348 dimensões. E o $LBP_{P1,R1}^{riu2} + LBP_{P2,R2}^{riu2} + LBP_{P3,R3}^{riu2}$ representado na tabela pela linha 3 RIU2_RIU2_RIU2 possui apenas 54 dimensões para o mesmo valor de pontos vizinhos $P1,P2,P3$.

Tabela 5 – Tamanho do vetor da combinação de 3 LBP

	Min	Max
	$P1=4,P2=4,P3=4$	$P1=16,P2=16,P3=16$
1. U2_U2_U2	45	729
2. RI_RI_RI	18	12348
3. RIU2_RIU2_RIU2	18	54
4. U2_RI_RIU2	27	4377
5. U2_U2_RI	36	4602
6. U2_U2_RIU2	36	504
7. U2_RI_RI	27	8475
8. U2_RIU2_RIU2	27	279
9. RI_RI_RIU2	18	8250
10. RI_RIU2_RIU2	18	4152

Fonte – O autor

5.2 Extração Simples com 1 Operador LBP

A extração com operador simples, ou extração sem combinar operadores, é apresentada em forma de diagrama como mostrado na Figura 19, que mostra o fluxo do método na busca do melhor operador LBP bem como dos melhores parâmetros de extração tendo a acurácia como medição de desempenho. Na parte esquerda da figura, temos as 350 imagens, que em seguida passa pela extração de características por cada um dos três operadores LBP, tendo seus parâmetros ponto e raio (P,R) modificados para obter extrações diferentes. Como foram usados 3 valores de pontos e 7 valores de raios, cada operador LBP simples extraiu 21 bases de dados referentes à todas combinações (P,R) possíveis. Cada uma das bases de dados passa, então, para a etapa de classificação, onde são separados o conjunto de treinamento e o de teste e classificados com SVM e KNN. As melhores acurácias de cada método são mostradas na Tabela 6.

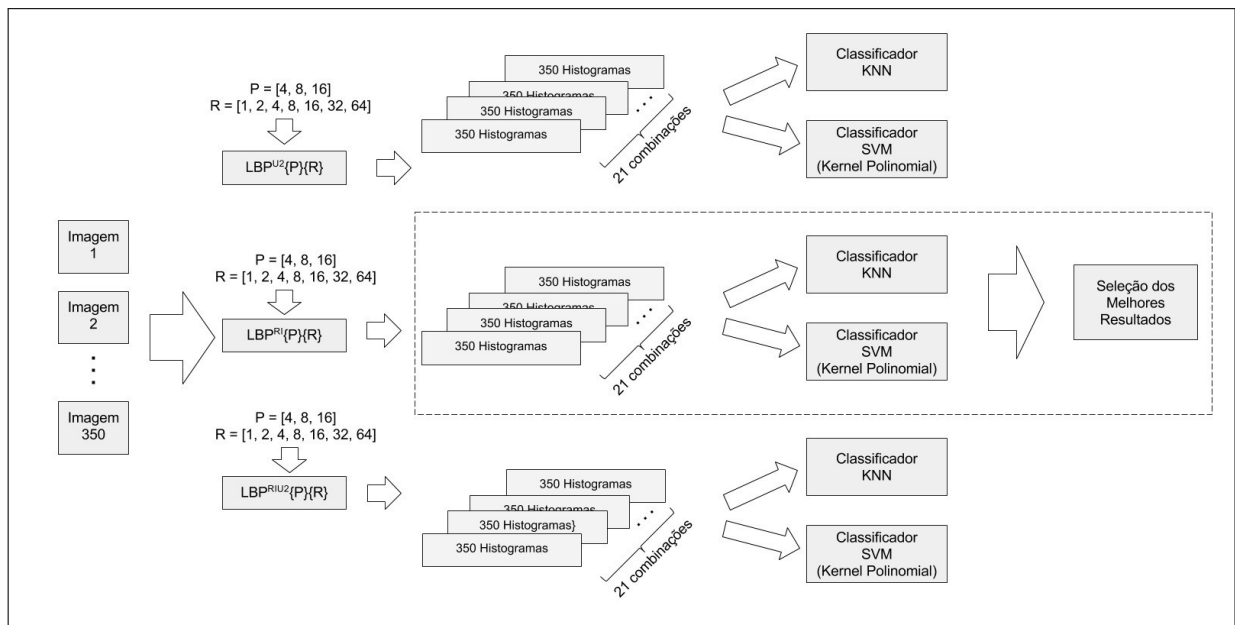
A Figura 20 mostra em um nível mais detalhado a estrutura da região pontilhada na Figura 19. A partir da obtenção dos histogramas, são feitos 5 ciclos de experimento utilizando diferentes amostras para treinamento e teste e, em seguida, treinados os classificadores KNN e SVM e calculada a acurácia média, depois é verificado qual conjunto de parâmetros LBP gerou a melhor acurácia média. Essa mesma estrutura foi também realizada de maneira análoga nos experimentos que envolvem a combinação de 2 e 3 métodos LBP.

Além do ajuste dos parâmetros de extração do LBP, os classificadores também tiveram seus parâmetros ajustados. No caso do KNN foram variados a quantidade k de vizinhos, e no caso do SVM, a variação foi feita modificando o grau p do polinômio do kernel.

O conjunto de valores avaliados para a variação dos parâmetros P , R , k e p foram os mesmos para as três abordagens e estão listados abaixo:

- Valores para a quantidade de pontos: $P \in \{4, 8, 16\}$;
- Valores para o raio: $R \in \{1, 2, 4, 8, 16, 32, 64\}$;
- Valor de k vizinhos no KNN: $k \in \{3, 5, 7\}$;
- Valor do grau do polinômio do kernel: $p \in \{1, 5, 10, 15, 20, 25, 30\}$.

Figura 19 – Diagrama da Extração com Combinação Simples



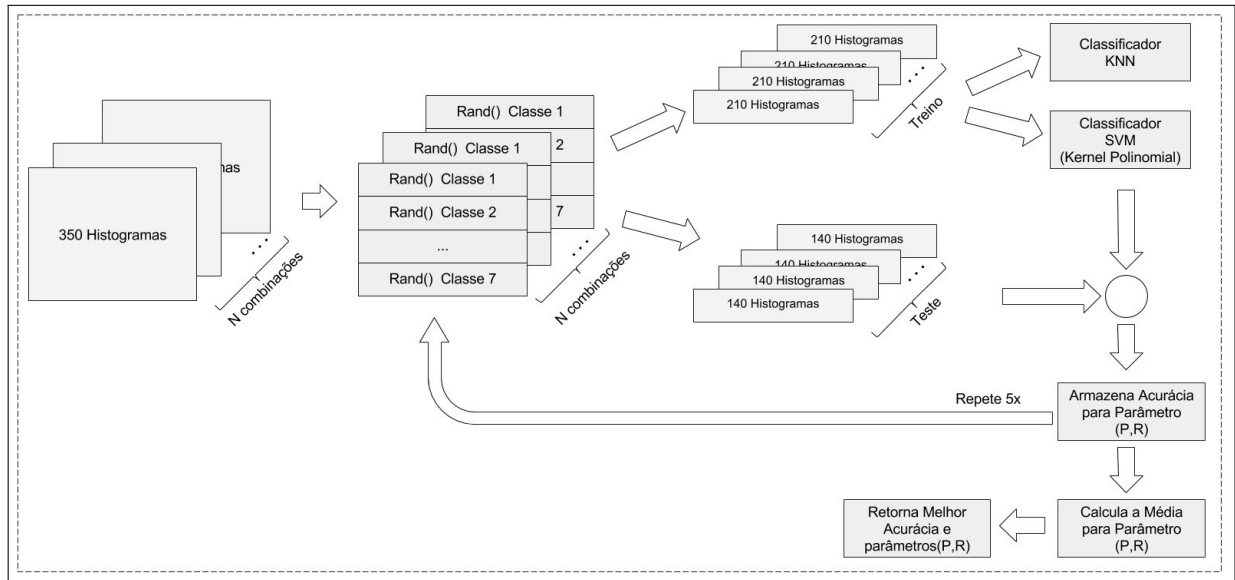
Fonte – O Autor

A Tabela 6 mostra os resultados da acurácia obtidos a partir da média da acurácia das classes. Podemos ver que o SVM foi melhor ou no mínimo, igual ao KNN. O melhor resultado foi obtido com o operador U2 e o RIU2, usando SVM e apresentando uma acurácia de 76,85%. Entretanto, estamos dando mais foco no comportamento do $LBP_{P,R}^{RIU2}$, que possui características mais adequadas para a aplicação do mundo real. O operador U2 é variante à rotação, ou seja, ele consegue capturar informação da angulação da imagem, sendo um efeito indesejável, pois não se pode garantir a angulação da peça de couro no momento da aquisição da imagem.

Portanto, para se garantir robustez na extração e melhores resultados, os operadores mais adequados são o RI e o RIU2, com uma ressalva para o operador RI, que é mais susceptível

a ruído e, além disso, pode retornar vetores com alta dimensionalidade dependendo dos valores de P e R escolhidos.

Figura 20 – Diagrama detalhado da região pontilhada da figura 19



Fonte – O Autor

Os valores dos parâmetros (P,R) para a obtenção de cada acurácia é mostrada no apêndice numa versão mais detalhada da Tabela.

Tabela 6 – Acurácia sem combinar LBP

Operador	KNN		SVM	
	Acurácia	σ	Acurácia	σ
1. U2	0.754286	0.0356	0.768571	0.0337
2. RI	0.741429	0.0421	0.745714	0.0415
3. RIU2	0.754286	0.0412	0.768571	0.0388

Fonte – O autor

Para uma análise mais detalhada do desempenho do operador RIU2 apresentamos a matriz de confusão para os classificadores KNN e SVM nas Tabelas 7 e 8.

Podemos ver que para o operador simples $LBP_{P,R}^{riu2}$ ambos os métodos tiveram comportamentos parecidos, possuindo a mesma tendência de erro, de diferentes maneiras. Ambos os métodos classificaram melhor a classe 6. A semelhança do comportamento também se verificou para a classe com pior desempenho, no caso a classe 5, em que pouco mais da metade foi classificada corretamente, com a mesma tendência de errar amostras da classe 5 como sendo a classe 2, 3, 4 ou 7. A tendência de erro também se verificou para as outras classes (2, 3 e 7),

Tabela 7 – Matriz de Confusão Operador Simples RIU2 com KNN

		Previsto						
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Real	Classe 1	14	1	2	0	0	0	3
	Classe 2	0	13	2	5	0	0	0
	Classe 3	3	0	13	1	0	1	2
	Classe 4	0	2	0	16	2	0	0
	Classe 5	0	3	1	4	11	0	1
	Classe 6	0	0	0	0	0	20	0
	Classe 7	2	2	2	0	1	0	13

Fonte – O autor

Tabela 8 – Matriz de Confusão Operador Simples RIU2 com SVM

		Previsto						
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Real	Classe 1	11	1	4	0	0	0	4
	Classe 2	0	14	1	5	0	0	0
	Classe 3	4	1	15	0	0	0	0
	Classe 4	0	2	0	16	2	0	0
	Classe 5	0	1	1	6	11	0	1
	Classe 6	0	0	0	0	0	20	0
	Classe 7	2	1	1	1	0	0	15

Fonte – O autor

tomando como exemplo a classe 2, que tanto o KNN quanto o SVM classificaram amostras da classe 2 como sendo da classe 3 ou 4, variando apenas na quantidade.

Vale lembrar que das diversas matrizes de confusão obtidas, foi mostrada apenas as matrizes referentes ao primeiro ciclo do experimento, que apesar de ser relativamente limitada, ilustra a tendência do comportamento de erros e acertos das classificações.

5.3 Extração Multi-resolução com 2 Operadores LBP

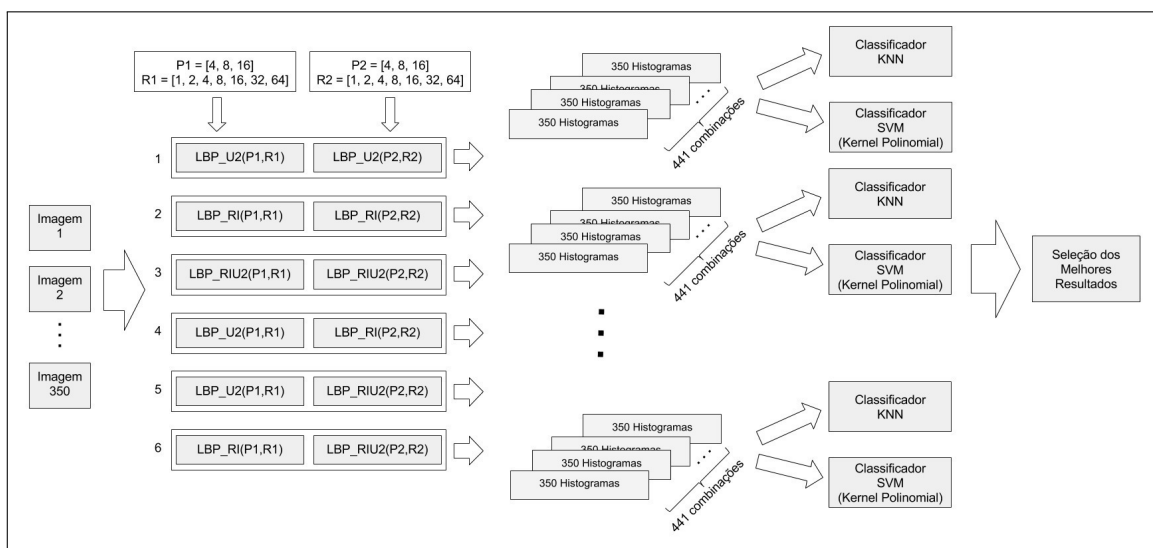
A extração de características com multi-resolução de dois operadores tem como objetivo obter mais informação da imagem, tendo vista que cada operador, mesmo que repetido, pode ter parâmetros (P,R) diferentes. A quantidade de pontos possui uma relação, não obrigatória, com o valor do raio, mas é importante ser observada para se evitar redundância na extração, principalmente quando se usa raios pequenos. Por exemplo, quanto menor o raio, menos pontos de vizinhança serão necessários para descrever a região da imagem. Muitos pontos para raio pequeno poderá trazer informação redundante. Por outro lado, quanto o raio, maior a quantidade pontos da vizinhança para se obter uma informação mais precisa da vizinhança. Para a invariância

à rotação, quanto maior a quantidade de pontos da vizinhança resolução angular, melhorando independência do método ao efeito da rotação.

A Figura 21 apresenta, em uma forma análoga a ilustração da combinação simples, como foi feita a extração para multi-resolução com dois operadores. Podemos ver que o número de bases de dados extraídas aumentou devido ao número de combinações possíveis de raio e pontos (P,R) de cada operador, desta vez temos dois pares (P1,R1) e (P2,R2). Lembrando que a combinação de operadores foi feita concatenando os vetores extraídos por cada operador. Note que na segunda etapa do processo foram geradas 441 base de dados para cada operadores (composto por dois operadores simples).

A tabela 9 apresenta os resultados obtidos para cada uma das seis combinações de dois operadores. Podemos observar que o SVM superou o KNN em todos os operadores. Mais uma vez é importante destacar que os operadores mais adequados para a aplicação são o $LBP_{P,R}^{RI}$ e/ou $LBP_{P,R}^{RIU2}$ bem como suas combinações, pois oferecem invariância à rotação. Portanto, o operador "2.RI_RI", "3.RIU2_RIU2" e o operador "6.RI_RIU2" são os mais interessantes para a aplicação, com uma ressalva de que o operador RI possui um grande número de dimensões conforme se aumenta número de pontos da vizinhança. Dado a pouca variação entre eles, a melhor escolha continua sendo o operador "3.RIU2_RIU2", pois é mais imune a ruído que o operador "2.RI_RI" e o "6.RI_RIU2", e possui menor dimensionalidade.

Figura 21 – Diagrama da Extração com Combinação Dupla



Fonte – O Autor

Quanto a análise da matriz de confusão do operador "3.RIU2_RIU2" percebemos

Tabela 9 – Melhor Acurácia média Combinando 2 Operadores LBP

Operador	KNN		SVM	
	Acurácia	σ	Acurácia	σ
1. U2_U2	0.800000	0.0431	0.847143	0.0433
2. RI_RI	0.767143	0.0442	0.831429	0.0341
3. RIU2_RIU2	0.784286	0.0313	0.830000	0.0389
4. U2_RI	0.784286	0.0439	0.845714	0.0260
5. U2_RIU2	0.791429	0.0427	0.841429	0.0358
6. RI_RIU2	0.780000	0.0424	0.830000	0.0340

Fonte – O autor

que tanto o KNN como o SVM continuam com a mesma tendência de erro para algumas classes como foi mostrado na extração sem combinar operadores. Por exemplo, amostras da classe 1 foram classificadas erroneamente com sendo das classes 3, 6 e 7 em ambos os classificadores, porém com menores erros no SVM. Amostras da classe 2 são classificadas erroneamente como sendo da classe 3 e 4 tanto no KNN como no SVM, contudo, o SVM inclui a classe 5. Note que o erro varia em quantidade, tendo o SVM melhor desempenho que o KNN.

O KNN teve melhor desempenho na classificação da classe 6, enquanto que o SVM teve melhor desempenho na classificação da classe 4. Outra observação que vale ser enfatizada é acurácia apresentada diz respeito a melhor escolha de parâmetros $(P1,R1)(P2,R2)$ e que são escolhidos de forma independente para o KNN e para o SVM. O que explicaria o melhor desempenho do KNN para a classe 6 e o melhor desempenho do SVM para a classe 4, pois a base de dados extraída para cada operador foi extraída com parâmetros diferentes. Nesse sentido pode-se pensar que não faz sentido comparar SVM e KNN, visto que utilizam bases de dados diferentes, porém o ponto de interesse se encontra na escolha dos melhores parâmetros de extração e no melhor classificador para os dados extraídos. A configuração dos parâmetros de ponto e raio dos operadores LBP são mostrados no apêndice em uma versão detalhada das tabelas, incluindo também o parâmetro k do KNN e o grau p do polinômio do *kernel* usado no SVM.

5.4 Extração Multi-resolução com 3 Operadores LBP

O processo da extração multi-resolução com três operadores LBP segue de maneira análoga às extrações anteriores, desta vez com 10 combinações utilizando três operadores simples. Cada uma das 10 combinações dos três de operador extraiu 9261 base de dados, referentes as diferentes combinações de $(P1,R1),(P2,R2)$ e $(P3,R3)$. A classificação com SVM foi feita

Tabela 10 – Matriz de Confusão Combinando 2 Operadores RIU2 com KNN

		Previsto						
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Real	Classe 1	13	0	3	0	0	1	3
	Classe 2	0	16	1	3	0	0	0
	Classe 3	3	2	15	0	0	0	0
	Classe 4	0	4	0	15	1	0	0
	Classe 5	0	2	0	6	12	0	0
	Classe 6	0	0	0	0	0	20	0
	Classe 7	5	0	0	0	1	0	14

Fonte – O autor

Tabela 11 – Matriz de Confusão Combinando 2 Operadores RIU2 com SVM

		Previsto						
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Real	Classe 1	17	0	1	0	0	1	1
	Classe 2	0	13	1	5	1	0	0
	Classe 3	3	2	14	0	0	0	1
	Classe 4	0	0	0	20	0	0	0
	Classe 5	0	0	0	4	16	0	0
	Classe 6	0	0	0	0	0	19	1
	Classe 7	7	0	0	0	0	0	13

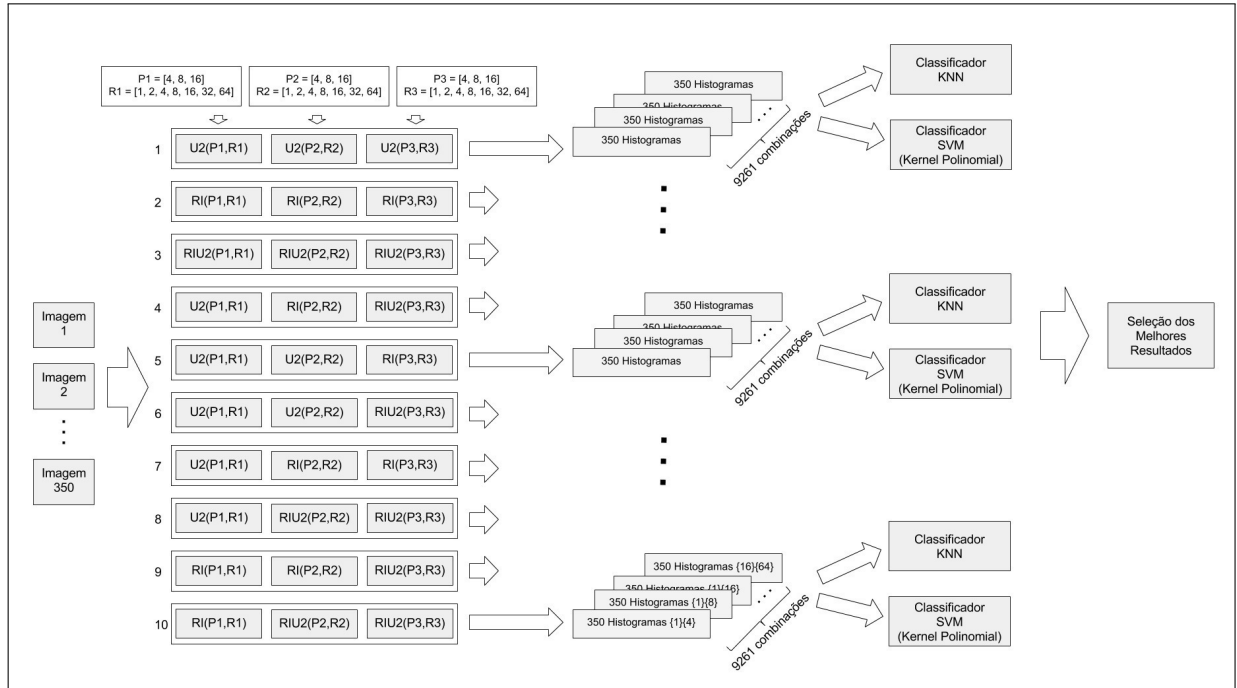
Fonte – O autor

apenas com um parâmetro para o *kernel* polinomial ($p=25$) devido a viabilidade de tempo de processamento.

O melhor resultado de acurácia média da tabela 12 mostra uma acurácia de 86%, mas devemos verificar que este operador, o U2_RI_RI (linha 8 da Tabela), apresenta um operador U2, que varia de acordo com a rotação, não sendo um resultado adequado para o tipo de aplicação, além de possuir dois operadores de alta dimensionalidade, no caso o RI. Como foi visto nos dois experimentos anteriores, o operador com menor dimensionalidade e maior tolerância a ruídos é o RIU2_RIU2_RIU2 (linha 3) que apresentou 83,42% de acurácia.

Analisando a matriz de confusão para o KNN e SVM, a tendência a errar classes se manteve, sendo que desta vez, os resultados da classe 3 foram mais distintos, o KNN errou amostras da classe 3 como sendo da classe 2 e 7, já o SVM errou amostras da classe 3 como sendo da classe 1 e 2. Ambos acertaram mais a classe 6 e ambos erraram mais a classe 1.

Figura 22 – Diagrama da Extração com Combinação Tripla



Fonte – Do Autor

Tabela 12 – Melhor Acurácia média Combinando 3 Operadores LBP

Operador	KNN		SVM	
	Acurácia	σ	Acurácia	σ
1. U2_U2_U2	0.801429	0.0386	0.831429	0.0256
2. RI_RI_RI	0.787143	0.0205	0.838571	0.0341
3. RIU2_RIU2_RIU2	0.790000	0.0293	0.834286	0.0234
4. U2_RI_RIU2	0.798571	0.0317	0.861429	0.0212
5. U2_U2_RI	0.797143	0.0345	0.852857	0.0265
6. U2_U2_RIU2	0.798571	0.0450	0.841429	0.0106
7. U2_RI_RI	0.792857	0.0303	0.860000	0.0260
8. U2_RIU2_RIU2	0.795714	0.0348	0.848571	0.0274
9. RI_RI_RIU2	0.790000	0.0179	0.840000	0.0284
10. RI_RIU2_RIU2	0.791429	0.0223	0.838571	0.0341

Fonte – O autor

Tabela 13 – Matriz de Confusão Combinando 3 Operadores RIU2 com KNN

	Classe 1	Classe 2	Classe 3	Previsto				
				Classe 4	Classe 5	Classe 6	Classe 7	
Real	Classe 1	12	0	5	0	0	1	2
	Classe 2	0	17	1	2	0	0	0
	Classe 3	0	1	16	0	0	0	3
	Classe 4	0	4	0	16	0	0	0
	Classe 5	0	5	0	2	13	0	0
	Classe 6	0	0	0	0	0	20	0
	Classe 7	3	0	1	0	1	1	14

Fonte – O autor

Tabela 14 – Matriz de Confusão Combinando 3 Operadores RIU2 com SVM

		Previsto						
		Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7
Real	Classe 1	14	0	4	0	0	1	1
	Classe 2	0	16	1	3	0	0	0
	Classe 3	2	3	15	0	0	0	0
	Classe 4	0	0	0	19	1	0	0
	Classe 5	0	1	0	3	16	0	0
	Classe 6	0	0	0	0	0	20	0
	Classe 7	3	0	0	0	1	0	16

Fonte – O autor

6 CONCLUSÕES E TRABALHOS FUTUROS

Dados os experimentos, podemos concluir que o LBP se mostrou capaz de discriminar os sete 7 tipos de classe de couro caprino. Foi visto também que apesar do extrator LBP uniforme ter obtido bons resultados, estudos teóricos mostram que a influência da rotação pode afetar o resultado da extração, sendo assim, para uso em aplicação prática é recomendável considerar o uso de métodos invariante à rotação.

Quanto a dimensionalidade, os extratores compostos pelo LBP invariante à rotação possuem grande dimensionalidade, afetando o tempo de processamento, além disso, são susceptíveis a ruídos, sendo esses dois fatores que limitam seu uso em aplicações do mundo real.

O Método LBP uniforme e invariante à rotação, apesar de ser o que extrai a menor quantidade de informação de textura, conseguiu descrever de maneira satisfatória as classes de qualidade de couro, com a vantagem de ser invariante à rotação, imunidade à ruídos e possuir baixa dimensionalidade.

Da extração com a combinação de métodos LBP com diferentes resoluções (P,R), esta foi capaz de melhorar a acurácia média dos modelos de classificação, porém para o caso onde temos 3 combinações envolvendo apenas o operador LBP RIU2 (invariante à rotação e uniforme), não houve relativa melhoria quando aumentada a quantidade de dois para três operadores, mantendo uma acurácia média próxima a 83%, conforme pode ser visto nas Tabelas 9 e 12.

Para trabalhos futuros, pode ser feito a seleção de atributos além da aplicação do LBP Multi-resolução com filtro passa-baixa (LBPF). A seleção de atributos visa investigar e escolher os melhores atributos de cada base gerada pela extração LBP com diferentes parâmetros (P,R), removendo componentes irrelevantes além da composição uma nova base usando atributos mistos (das bases de diferentes extrações). Em tese, a seleção pode melhorar o desempenho da classificação além de reduzir a dimensionalidade na composição de bases. Diversos métodos poderão ser aplicados, como *Sequential Feature Selection* (WHITNEY, 1971), ou o *Floating Search Methods* (PUDIL *et al.*, 1994). Outros métodos de seleção de atributos estão estruturados e reunidos em (SHEIKHPOUR *et al.*, 2017).

Já no LBPF, a multi-resolução é feita aplicando diferentes filtros gaussianos para cada distância de raio adotado de modo que os pontos amostrados por raios maiores representem melhor a área de abrangência daquele raio, complementando a informação de textura (MÄENPÄÄ, 2003).

REFERÊNCIAS

- AHONEN, T.; MATAS, J.; HE, C.; PIETIKÄINEN, M. Rotation invariant image description with local binary pattern histogram fourier features. In: SPRINGER. **Scandinavian Conference on Image Analysis**. [S.l.], 2009. p. 61–70.
- AMORIM, W. P. **Redução de atributos utilizando análise discriminante com aplicações na detecção de defeitos em couro bovino**. Dissertação (Mestrado), 2009.
- BISHOP, C. M. Machine learning and pattern recognition. **Information Science and Statistics**. Springer, Heidelberg, 2006.
- BRODATZ, P. **Textures: A Photographic Album for Artists and Designers**. Dover Publications, 1999. Disponível em: <<http://www.ux.uis.no/~tranden/brodatz.html>>.
- BRUNELLI, R.; POGGIO, T. Face recognition: Features versus templates. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 15, n. 10, p. 1042–1052, 1993.
- GOMES, J. P. P. **Reconhecimento Automático de Alvos em Imagens SAR**. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006.
- HARALICK, R. M.; SHANMUGAM, K. *et al.* Textural features for image classification. **IEEE Transactions on systems, man, and cybernetics**, Ieee, v. 3, n. 6, p. 610–621, 1973.
- LIU, H.; MOTODA, H. **Feature extraction, construction and selection: A data mining perspective**. [S.l.]: Springer Science & Business Media, 1998. v. 453.
- LIU, L.; FIEGUTH, P.; GUO, Y.; WANG, X.; PIETIKÄINEN, M. Local binary features for texture classification: Taxonomy and experimental study. **Pattern Recognition**, Elsevier, v. 62, p. 135–160, 2017.
- LIU, L.; ZHAO, L.; LONG, Y.; KUANG, G.; FIEGUTH, P. Extended local binary patterns for texture classification. **Image and Vision Computing**, Elsevier, v. 30, n. 2, p. 86–99, 2012.
- MÄENPÄÄ, T. **The local binary pattern approach to texture analysis: extensions and applications**. Tese (Doutorado), 2003. Disponível em: <<http://herkules.oulu.fi/isbn9514270762/>>.
- MITCHELL, T. **Machine Learning**. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: <<https://books.google.com.br/books?id=EoYBngEACAAJ>>.
- NG, A. Y.; JORDAN, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2002. p. 841–848.
- NIXON, M. S.; AGUADO, A. S. **Feature extraction & image processing for computer vision**. [S.l.]: Academic Press, 2012.
- OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: **IEEE. Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing, Proceedings of the 12th IAPR International Conference on**. [S.l.], 1994. v. 1, p. 582–585.

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.

PETROU, M.; SEVILLA, P. G. **Image processing: dealing with texture**. [S.l.]: Wiley Online Library, 2006. v. 1.

PUDIL, P.; NOVOVIČOVÁ, J.; KITTLER, J. Floating search methods in feature selection. **Pattern recognition letters**, Elsevier, v. 15, n. 11, p. 1119–1125, 1994.

ROCHA NETO, A. R.; SOUSA, R.; BARRETO, G. de A.; CARDOSO, J. Diagnostic of pathology on the vertebral column with embedded reject option. **Pattern Recognition and Image Analysis**, Springer, p. 588–595, 2011.

SANTOS FILHO, E. Q. d. **Um Sistema de visão computacional para classificação da qualidade do couro caprino**. Dissertação (Mestrado), 2013.

SHEIKHPOUR, R.; SARRAM, M. A.; GHARAGHANI, S.; CHAHOOKI, M. A. Z. A survey on semi-supervised feature selection methods. **Pattern Recognition**, Elsevier, v. 64, p. 141–158, 2017.

STATNIKOV, A.; HARDIN, D.; GUYON, I.; ALIFERIS, C. F. A gentle introduction to support vector machines in biomedicine. 2011.

WANG, L.; HE, D. A new statistical approach for texture analysis. **Photogrammetric Engineering and Remote Sensing**, v. 56, n. 1, p. 61–66, 1990.

WANG, L.; HE, D.-C. Texture classification using texture spectrum. **Pattern Recognition**, Elsevier, v. 23, n. 8, p. 905–910, 1990.

WHITNEY, A. W. A direct method of nonparametric measurement selection. **IEEE Transactions on Computers**, IEEE, v. 100, n. 9, p. 1100–1103, 1971.

7 RESULTADOS COM CONFIGURAÇÃO DOS PARÂMETROS LBP

A seguir apresentamos a configuração dos parâmetros (P,R) de cada LBP para a obtenção da acurácia mostrada nas tabelas dos resultados. Note que a melhor acurácia para o SVM não apresenta os mesmos parâmetros (P,R) da melhor acurácia do KNN.

Tabela 15 – Melhor Acurácia média sem combinar LBP - Detalhada

	KNN			SVM		
	(P,R),k	Acurácia	σ	(P,R),p	Acurácia	σ
1. U2	(8,4),5	0.754286	0.0356	(8,64),30	0.768571	0.0337
2. RI	(8,4),7	0.741429	0.0421	(4,2),30	0.745714	0.0415
3. RIU2	(8,2),3	0.754286	0.0412	(16,8),30	0.764286	0.0388

Fonte – O autor

Tabela 16 – Melhor Acurácia média Combinando 2 Operadores LBP - Detalhada

	KNN			SVM		
	(P1,R1)(P2,R2),k	Acurácia	σ	(P1,R1)(P2,R2),p	Acurácia	σ
1. U2_U2	(8,4),(8,64),3	0.800000	0.0431	(4,1),(8,16),20	0.847143	0.0433
2. RI_RI	(4,32),(8,1),7	0.767143	0.0442	(8,8),(8,32),25	0.831429	0.0341
3. RIU2_RIU2	(8,1),(16,64),5	0.784286	0.0313	(8,16),(16,1),20	0.830000	0.0389
4. U2_RI	(8,4),(16,32),3	0.784286	0.0439	(8,16),(16,1),20	0.845714	0.0260
5. U2_RIU2	(8,64),(8,4),3	0.791429	0.0427	(8,16),(8,1),20	0.841429	0.0358
6. RI_RIU2	(16,32),(8,4),7	0.780000	0.0424	(8,32),(16,4),30	0.830000	0.0340

Fonte – O autor

Tabela 17 – Melhor Acurácia média Combinando 3 Operadores LBP - Detalhada

	KNN			SVM		
	(P1,R1)(P2,R2)(P3,R3),k	Ac.	σ	(P1,R1)(P2,R2)(P3,R3),p	Ac.	σ
1. U2_U2_U2	(8,4),(8,4),(8,64),3	0.8014	0.0386	(8,4),(16,16),(16,64),25	0.8314	0.0256
2. RI_RI_RI	(8,2),(16,16),(16,64),3	0.7871	0.0205	(8,1),(8,16),(8,32),25	0.8386	0.0341
3. RIU2_RIU2_RIU2	(4,1),(16,4),(16,64),5	0.7900	0.0293	(4,1),(8,32),(16,16),25	0.8343	0.0234
4. U2_RI_RIU2	(8,64),(16,32),(8,4),3	0.7986	0.0317	(16,64),(8,32),(16,4),25	0.8614	0.0212
5. U2_U2_RI	(8,4),(8,64),(16,16),3	0.7971	0.0345	(16,4),(16,64),(8,32),25	0.8528	0.0265
6. U2_U2_RIU2	(8,4),(8,64),(16,4),3	0.7986	0.0450	(8,4),(16,64),(16,64),25	0.8414	0.0106
7. U2_RI_RI	(8,64),(8,1),(8,16),3	0.7929	0.0303	(16,64),(8,32),(16,4),25	0.8600	0.0260
8. U2_RIU2_RIU2	(8,64),(8,16),(16,4),3	0.7957	0.0348	(16,64),(8,32),(16,8),25	0.8486	0.0274
9. RI_RI_RIU2	(16,2),(16,64),(16,32),7	0.7900	0.0179	(8,1),(8,32),(8,16),25	0.8400	0.0284
10. RI_RIU2_RIU2	(16,1),(16,4),(16,64),5	0.7914	0.0223	(8,32),(8,1),(8,16),25	0.8385	0.0341

Fonte – O autor

8 TEMPO DE PROCESSAMENTO

Quanto ao tempo aproximado de processamento para um ciclo de experimentos estão dispostos abaixo, para um sistema composto por processador core i5 5200U, 2.2GHz, 4GB RAM.

8.1 Tempo do classificador simples

- 3 combinações de LBP – KNN e SVM
- 63 bases x [(3 parâmetros KNN) + (7 parâmetros SVM)] = 630 resultados
- 4 minutos

8.2 Tempo do classificador com 2 combinações

- 6 combinações de LBP - KNN e SVM
- 2.646 bases x [(3 parâmetros KNN) + (7 parâmetros SVM)] = 26460 resultados
- 4 Horas e 30 minutos

8.3 Tempo do Classificador com 3 combinações

- 10 combinações de LBP – KNN e SVM
- 92.610 bases x [(3 parâmetros KNN) + (1 parâmetros SVM)] = 370440 resultados
- 55 Horas