



Pós-Graduação em Ciência da Computação

MARRONE SILVÉRIO MELO DANTAS

**Classificação de texturas baseada em quantização
de vetores com diferença local**



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
<http://cin.ufpe.br/~posgraduacao>

Recife
2019

MARRONE SILVÉRIO MELO DANTAS

**Classificação de texturas baseada em quantização
de vetores com diferença local**

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Visão Computacional e Aprendizagem de Máquinas.

Orientador: Tsang Ing Ren

Coorientador: Tiago Buarque Assunção de Carvalho

Recife
2019

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

D192c Dantas, Marrone Silvério Melo
Classificação de texturas baseada em quantização de vetores com
diferença local / Marrone Silvério Melo Dantas. – 2019.
73 f.: il., fig., tab.

Orientador: Tsang Ing Ren.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,
Ciência da Computação, Recife, 2019.
Inclui referências.

1. Visão computacional. 2. Aprendizagem de máquina. 3. Quantização de
vetores. I. Ren, Tsang Ing (orientador). II. Título.

006.37 CDD (23. ed.) UFPE- MEI 2019-059

MARRONE SILVÉRIO MELO DANTAS

“Classificação de Texturas Baseada em Quantização de Vetores com Diferença Local”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 19/02/2019.

Orientador: Prof. Dr. Tsang Ing Ren

BANCA EXAMINADORA

Prof. Dr. Carlos Alexandre Barros de Mello (Examinador Interno)
Centro de Informática / UFPE

Prof. Dr. Francisco Madeiro Bernardino Junior (Examinador Externo)
Centro de Ciências e Tecnologia / UNICAP

AGRADECIMENTOS

Ao meu orientador, Professor Dr. Tsang Ing Ren, por todo o ensino no decorrer do mestrado, a confiança depositada em mim e toda a paciência.

Ao meu coorientador, Professor Dr. Tiago Buarque Assunção de Carvalho, por todo o auxílio e paciência no desenvolvimento desse trabalho.

À minha mãe Silvia Roseane, uma grande inspiração, por todo o apoio não só nesse processo como na minha vida. Aos meus irmãos André e Maria que me apoiaram em tudo com conselhos e amor.

Ao meu pai João Dantas, que mesmo não estando entre nós, encaminhou-me para uma vida de estudos e aprendizagem.

Aos meus tios, Fátima, Osvaldo e Flávio, que além de acolhimento, forneceram-me conselhos, harmonia e compreensão necessárias para o meu crescimento e conclusão desse trabalho.

À minha avó Rosa Dantas que fomentou e me incentivou em todas as fases do meu estudo, pois sem ela o início dessa jornada não seria possível.

Ao CNPq, pelo apoio financeiro no mestrado.

A todos os amigos e familiares que de alguma forma contribuíram para conclusão desta dissertação.

RESUMO

Local Binary Pattern (LBP) é um simples e poderoso descritor visual utilizado para classificação de texturas e diversas outras aplicações em visão computacional. Apesar do LBP ser uma poderosa ferramenta ele pode apresentar algumas limitações, já que seu processamento pode gerar o mesmo código para padrões de texturas diferentes, além de possuir uma fraca robustez a diversos tipos de ruído. O *Local Vector Quantization Pattern* (LVQP) é uma extensão do LBP que tem como proposta lidar com essas limitações. Neste trabalho propomos o método *Difference Quantization Pattern* (DQP), que é a utilização do vetor de diferenças em conjunto com a técnica de clusterização conhecida como quantização de vetores, e o combinamos com o LVQP para propor o D-LVQP. Para o D-LVQP, os histogramas resultantes do DQP e LVQP são concatenados e usados como características para um classificador k -NN. O método proposto foi aplicado em cinco bases: UIUC, CURET, KTH-TIPS, KTH-TIPS2 e BRODATZ. D-LVQP obteve uma acurácia superior em todos os casos quando comparado ao LVQP, demonstrando assim uma maior robustez.

Palavras-chaves: LBP. LVQP. Quantização de Vetores. Classificação de Texturas.

ABSTRACT

Local Binary Pattern (LBP) is a simple and powerful visual descriptor most used in texture classification but can be applied and several other applications in Computer Vision. However, LBP presents some limitations since its possible to generate the same code for different texture patterns, and it has a weak resistance to noise. Local Vector Quantization Pattern (LVQP), which is an extension of LBP, was proposed to solve this kind of issues. Here, we propose an alternative descriptor based on the difference in the clustering method vector quantization, Difference Quantization Pattern (DQP) and a combined DQP and LVQP descriptor named Difference-Local Vector Quantization Pattern (D-LVQP). For the D-LVQP method, the histograms from DQP and LVQP are concatenated and used as features for the KNN classifier. We applied the proposed method to five datasets UIUC, CURET, KTH-TIPS, KTH-TIPS2, and BRODATZ. D-LVQP obtained a superior accuracy in most cases when compared to LVQP, thus demonstrating greater robustness.

Keywords: LBP. LVQP. Vector Quantization. Texture Classification.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de uma aplicação de rotação em uma imagem de textura. Em (a) temos a imagem original sem rotação e em (b) o resultado da rotação quando a aplicamos com o valor de ângulo θ de 90 graus. . . .	18
Figura 2 – Exemplo de redução da imagem da textura. Na Figura (a) temos a textura original, já na Figura (b) temos a mesma imagem com suas dimensões reduzidas em 50%	18
Figura 3 – Exemplo de aplicação de ruído aleatório em uma imagem de textura. Na Figura (a) temos a imagem original sem ruído, na Figura (b) temos a imagem com ruído aleatório aplicado	19
Figura 4 – Exemplo de aplicações com textura: Em (a) temos um conjunto de cinco texturas distintas. Uma aplicação possível seria a classificação em que cada textura receberia uma rótulo. Em (b). Um segundo caso possível seria a segmentação, na qual para cada tipo de textura seria definida uma região delimitante (c)	24
Figura 5 – Exemplo da aplicação dos descritores de Haralick. Primeiro temos a textura original (a), depois geramos a matriz de correlação entre os pixels e seus vizinhos, no caso (b) é computada a relação com o vizinho a direita foi o considerado, depois a matriz é normalizada entre 0 e 1 (c) e como passo final são computados os descritores (d).	28
Figura 6 – Exemplo da aplicação dos GLCM. Primeiro temos a textura original (a), depois selecionamos os descritor utilizados no GLCM (b), geramos as matrizes de correlação e efetuamos a extração dos descritores (c) e finalmente obtemos as novas caraterísticas da textura (d).	28
Figura 7 – Exemplo da estrutura de um TU 3x3. Cada célula representa a intensidade de um pixel, sendo o V_0 o pixel central, e todos as outras células que o circundam, seus vizinhos, com seus respectivos valores de intensidades.	29
Figura 8 – Exemplo aplicações da codificação TU. Em (8a) temos o bloco de pixels de exemplo com o valor do pixel central $V_0 = 70$, em (8b) temos o bloco de pixels processados com os novos valores componentes do vetor de características E	30
Figura 9 – Fluxograma da abordagem TU. Primeiro a partir da textura (a) aplicamos a codificação TU (b); em seguida, para cada código aplicamos a codificação <i>Unit Number</i> (c), logo após extraímos o histograma da textura (d), e, por final, obtemos o histograma como vetor final de característica. (e)	30

Figura 10 – Exemplo da codificação LBP original. Primeiro, temos o bloco de textura original (a); em seguida, aplicamos a limiarização nos vizinhos (b) e multiplicamos pelos pesos já definidos (c) e obtemos os novos valores da codificação (d).	31
Figura 11 – Exemplo da aplicação da codificação LBP. Primeiro temos como entrada uma textura original (a) sem codificação; em seguida, aplicamos a codificação LBP na textura (b) e o histograma da textura codificada é calculado (c).	33
Figura 12 – Exemplo de abordagem LBP com variação de raio e total de vizinhos.	33
Figura 13 – Exemplo da aplicação da interpolação bilinear, considerando os vizinhos próximos nas posições $(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2)$, seus valores de intensidade respectivamente $(Q_{11}, Q_{12}, Q_{21}, Q_{22})$ e, por fim, o novo valor de pixel p nas posições (x, y) definidos pela interpolação bilinear.	34
Figura 14 – Exemplo do processo da geração do código LBP a partir de um bloco de pixels e de um novo valor do pixel central. Primeiro temos o bloco de pixel original (a) então obtemos os valores da binarização dos vizinhos (b) a partir da binarização obtemos uma vetor binários (c) que como saída é convertido para base 10 se tornando o novo código LBP (d) . .	35
Figura 15 – Fluxograma geral do LBP. Primeiro selecionamos a textura original (a), então a imagem é codificada pelo LBP (b) que tem como saída a imagem codificada (c); da imagem codificada, é então extraído o histograma (d) que é utilizado como conjunto de característica para algum algoritmo de classificação(e).	36
Figura 16 – Exemplo da aplicação do invariante de rotação com transições espaciais. No exemplo temos dois códigos diferentes, em (a) temos um código uniforme, ou seja, é uniforme, retornando a soma do vetor binário igual a 5, no segundo caso (b) temos 4 transições espaciais, ou seja, não é uniforme, retornando o valor de $P + 1 = 8 + 1 = 9$	37
Figura 17 – Busca no <i>codebook</i> e associação no novo código LVQP. O primeiro passo é a seleção de uma textura (a), então um bloco de pixels é extraído (b); com o bloco selecionado é extraído o vetor de diferença (c); com o vetor de diferença adquirido é efetuada uma busca no <i>codebook</i> pela menor distância entre o vetor de diferença e os centróides componentes do codebook (d) e, finalmente, o código resultante é a posição do centróide no <i>codebook</i> , no exemplo acima é o código 37.	43
Figura 18 – Exemplo de rotação de uma código com dimensão 8. O código na posição 1 é o código original obtido do vetor de diferença, os códigos seguintes são resultados de rotação.	43

Figura 19 – LVQP x LBP. Aqui podemos observar que dois padrões diferentes produzem a mesma codificação LBP enquanto produzem codificações LVQP diferentes.	44
Figura 20 – Diagrama simplificado do método DQP. O processo é dividido basicamente em 4 passos: criação da base de treino com texturas de uma determinada base (a), extração de todos os vetores de diferença (b), união dos vetores de diferenças de forma que os códigos gerados tenham dimensão 1 (c) e, por fim, o envio da base gerada para a quantização de vetores (d).	45
Figura 21 – Exemplo de processo de busca DQP. Primeiro iniciamos o processo com a extração do bloco de pixels de uma textura (a). A partir de um bloco de pixel, extraímos o vetor diferenças (b). Com o vetor de diferença, obtemos todas as distâncias do código gerado entre todos os componentes do <i>codebook</i> (c). Repetimos o processo para todos os códigos gerados do vetor de diferença (d) e, a partir do centróide com a menor distância encontrada dentre todos os códigos gerados do vetor de diferença, extraímos a sua posição no <i>codebook</i> , que é utilizada como um código DQP (e), substituindo o valor de intensidade do pixel em matriz nova na mesma posição que o pixel central original.	45
Figura 22 – Geração de histograma após abordagem LVQP. Na figura (a), temos primeiramente um conjunto de imagens com dimensão arbitrária Altura (A) e Largura (L) , cada imagem tem AL características. Quando for aplicado na abordagem LVQP com 8 clusters (b), como saída teremos 8 níveis de intensidades que servem como dado para a geração dos histograma que possui igualmente 8 níveis (c).	46
Figura 23 – Diagrama do método proposto (D-LVQP). Seguindo o exemplo, primeiro extraímos a base em treino e teste. A base de treino tem dois <i>codebooks</i> : um para o DQP e outro LVQP. A base de teste é convertida em duas codificações DQP e LVQP, das imagens de saída. De cada uma delas é computado o histograma DQP e LVQP para, finalmente, ser gerado o histograma D-LVQP(DQP+LVQP).	47
Figura 24 – Exemplo das 25 classes da base UIUC	49
Figura 25 – Exemplos das bases da base CuRet	52
Figura 26 – Exemplo das 10 classes da base KTH-TIPS	54
Figura 27 – Exemplos das 11 classes da base KTH-TIPS2b	57
Figura 28 – Exemplos das classes da base Brodatz	59

LISTA DE TABELAS

Tabela 2 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).	50
Tabela 3 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).	50
Tabela 4 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).	51
Tabela 5 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).	52
Tabela 6 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).	53
Tabela 7 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).	54
Tabela 8 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).	55
Tabela 9 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).	55
Tabela 10 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).	56
Tabela 11 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).	57
Tabela 12 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).	58
Tabela 13 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).	58
Tabela 14 – Classificação em relação à acurácia (%) para a base Brodatz com diferentes métodos	60
Tabela 15 – Comparação em relação à acurácia de métodos de aprendizagem profunda para classificação de texturas em relação ao método proposto	61
Tabela 16 – Comparação do método LVQP em relação o método D-LVQP para a base UIUC para o T-test de Welch	63
Tabela 17 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 5$ para a base UIUC	63
Tabela 18 – Comparação do método LVQP em relação o método D-LVQP para a base CuRet para o T-test de Welch	64

Tabela 19 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 6$ para a base CuRet	64
Tabela 20 – Comparação do método LVQP em relação o método D-LVQP para a base KTH-TIPS para o T-test de Welch	65
Tabela 21 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 5$ para a base UIUC	65
Tabela 22 – Comparação do método LVQP em relação ao método D-LVQP para a base KTH-TIPS2B para o T-test de Welch	65
Tabela 23 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 7$ para a base KTH-TIPS2b	66
Tabela 24 – Tabela de ganho do método D-LVQP em relação ao LVQP para a base Brodatz	67

LISTA DE ABREVIATURAS E SIGLAS

BRINT	<i>Binary Rotation Invariant</i>
CLBP	<i>Complete Local Binary Pattern</i>
DLBP	<i>Dominant LBP</i>
D-LVQP	<i>Difference-Local Vector Quantization Pattern</i>
DQP	<i>Difference Quantization Pattern</i>
ELBP	<i>Extended Local Binary Pattern</i>
GLCM	<i>Level Co-occurrence Matrix</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
JEMDLBP	<i>Joint Encoding of Multi-direction LBP</i>
LBP	<i>Local Binary Pattern</i>
LTP	<i>Local Ternary Pattern</i>
LVQP	<i>Local Vector Quantization Pattern</i>
MRF	<i>Markov Random Fields</i>
TU	<i>Text Unit</i>

LISTA DE SÍMBOLOS

x	Posição horizontal de uma matriz
y	Posição vertical de uma matriz
x'	Nova posição horizontal de uma matriz
y'	Nova posição vertical de uma matriz
$\rightarrow$	Transformação girométrica
θ	Transformação girométrica
X	Altura da matriz ou imagem
Y	Altura da matriz ou imagem
S_x	Fator de expansão em relação a X
S_y	Fator de expansão em relação a Y
Σ	Somatório
P	Total de vizinhos selecionados
R	Dimensão do raio
V_d	Vetor de diferença
K	Total de <i>clusters</i>
ϵ	Distorção máxima
N	Número de imagens por treino

SUMÁRIO

1	INTRODUÇÃO	16
1.1	MOTIVAÇÃO	17
1.2	OBJETIVOS	20
1.3	ESTRUTURA DO TRABALHO	20
2	ANÁLISE DE TEXTURAS	22
2.1	DEFINIÇÃO DE TEXTURA	22
2.2	DESCRIÇÃO DE TEXTURAS	24
2.2.1	Método Estatístico	24
2.2.2	Método Estrutural	25
2.2.3	Método Baseado em Modelo	26
2.3	DESCRITORES DE HARALICK	26
2.4	O LOCAL BINARY PATTERN - LBP	29
2.5	PRIMEIRA ABORDAGEM LBP	31
2.6	LBP - NOVO MODELO	32
2.7	INVARIANTE DE ROTAÇÃO UNIFORME	36
2.8	VARIANTES DO LBP	37
3	MÉTODO PROPOSTO	40
3.1	LOCAL VECTOR QUANTIZATION PATTERN - LVQP	40
3.2	VETOR DE DIFERENÇA	40
3.3	QUANTIZAÇÃO DE VETORES	41
3.4	ALGORITMO - LVQP	42
3.5	DIFFERENCE QUANTIZATION PATTERN - DQP	44
3.6	DIFFERENCE-LVQP - (D-LVQP)	46
4	RESULTADOS EXPERIMENTAIS	48
4.1	VARIÁVEIS, DEFINIÇÕES E MEDIDAS	48
4.2	BASE UIUC	49
4.3	BASE CURET	51
4.4	BASE KTH-TIPS	54
4.5	BASE KTH-TIPS2B	56
4.6	BASE BRODATZ	59
4.7	MÉTODOS LBP X APRENDIZAGEM PROFUNDA	60
4.8	AVALIAÇÃO DOS RESULTADOS	62
4.9	BASE UIUC	62

4.10	BASE CURET	63
4.11	BASE KTH-TIPS	64
4.12	BASE KTH-TIPS2B	65
4.13	BASE BRODATZ	67
5	CONCLUSÕES E TRABALHOS FUTUROS	68
	REFERÊNCIAS	70

1 INTRODUÇÃO

A visão é um dos principais e mais úteis dos sentidos humanos. A compreensão do mundo e dos seus componentes se dá de forma mais clara e abrangente com o auxílio da visão. O campo da ciência ligada à computação que tenta interpretar a capacidade da visão humana para uma linguagem de máquina é denominado Visão Computacional. A Visão Computacional demonstrou ser útil em diversas áreas como citado por Sze Lisk (2010), tais como reconhecimento de caracteres, inspeção de máquinas, modelagem 3D, reconhecimento de faces, entre outras.

Das diversas aplicações que podem ser desenvolvidas com visão computacional, algumas das mais relevantes são as que tentam solucionar problemas relacionados à textura. A definição de textura ainda é bastante divergente, mas de forma geral existe uma concordância quanto a ela ser uma informação de baixo nível de uma imagem, valendo dizer, portanto, que problemas de classificação da textura estão entre os mais relevantes. A classificação de texturas pode ser útil nas mais diversas áreas; como menciona Ito (2009) ao dizer que ela é utilizada na inspeção industrial, biomédica, na classificação e análise de imagens de sensoriamento remoto e busca e identificação por imagens com determinadas características num banco de dados. Aplicações importantes da classificação de texturas incluem diversas variantes desde a área médica, com detecção de câncer (BABAGHORBANI *et al.*, 2010); até a área agrícola, com a detecção de qualidade de frutas (GARCIA *et al.*, 2016).

Para a classificação de texturas, diversas abordagens foram criadas, a maioria delas baseada em descritores visuais. Um dos mais famosos descritores é o *Local Binary Pattern* (LBP) (OJALA; PIETIKAINEN; HARWOOD, 1994). As abordagens baseadas em LBP extraem as características das imagens codificadas e geram um histograma como um novo vetor de características para ser utilizado na classificação. O LBP, apesar de ser rápido e robusto para vários casos, ainda pode apresentar algumas deficiências relacionadas à rotação, ao ruído e à iluminação. Diversas variantes foram propostas para solucionar esses problemas, uma delas sendo o *Local Vector Quantization Pattern* (LVQP) (PAN; FAN; ZHANG, 2015). O LVQP é um método eficiente, porém ainda possui algumas limitações, das quais podemos citar duas: dependência de uma alta quantidade de dados para treino e uma possível perda de informações relevantes ocasionada pela redução de dimensionalidade. O objetivo desse trabalho é demonstrar que o método proposto *Difference-Local Vector Quantization Pattern* (D-LVQP) possui um desempenho superior em relação à acurácia quando comparado ao método de origem LVQP e diversos outros descritores locais, sendo esse ganho uma consequência da resolução das limitações do LVQP.

1.1 MOTIVAÇÃO

Existem diversas abordagens para a classificação de texturas; elas, em sua maioria, baseiam-se em extração de características em conjunto com um classificador. Extrair informação da textura em si é um processo difícil e complicado, embora haja uma abordagem possível para a extração de dados relevantes, como é o caso da utilização de um descritor. Um descritor é uma técnica que descreve características de uma imagem (cor, forma, textura, etc), de tal forma que imagens que apresentem características similares sejam descritas da forma mais semelhante possível. De um descritor é esperado que seus dados sejam relevantes com o mínimo de ambiguidades entre eles. Alguns dos fatores que podem dificultar a descrição são: rotação, mudança de escala e presença de ruído.

Das condições citadas, a rotação e mudança de escala são modificações geométricas, e para entendê-las primeiro devemos considerar uma imagem como uma matriz, cada posição da matriz (x, y) se refere a um valor de pixel. Uma modificação geométrica, como pode ser vista na Equação (1.1) consiste em dado um conjunto de coordenadas, referenciando um pixel na imagem original, é gerado como saída uma nova posição desse pixel em uma nova imagem:

$$I(x, y) \rightarrow I'(x', y'). \quad (1.1)$$

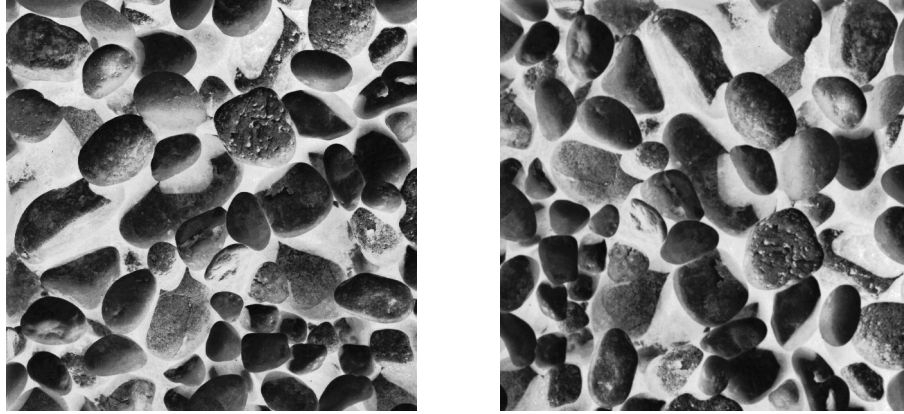
Na Equação (1.1) temos que I é a imagem original e (x, y) coordenadas originais de um pixel que, após a transformação geométrica ($\rightarrow$), a intensidade desse pixel assume uma nova posição (x', y') na nova imagem I' . Uma das mais comuns transformações é a rotação, na qual, a partir de um ângulo θ , a imagem gira a partir de um ponto. Podemos observar nas Equações (1.2) que obtemos novas coordenadas do pixel da imagem original a partir de uma relação com o ângulo θ .

$$x' = x.\cos(\theta) - y.\sin(\theta). \quad (1.2a)$$

$$y' = x.\sin(\theta) + y.\cos(\theta). \quad (1.2b)$$

Nas Equações (1.2) temos que (x', y') são as novas coordenadas, (x, y) as coordenadas da imagem original e θ o ângulo de rotação. Na Equação (1.2a) temos a nova posição de x , seja x' em relação a origem do plano cartesiano ($x = 0, y = 0$) a partir de um ângulo θ ; de forma similar temos a mesma aplicação para y' como a nova posição de y na Equação (1.2b). Como resultado de uma rotação, temos a Figura 1, na figura da esquerda (a), temos a imagem original e na figura da direita (b), temos a imagem rotacionada em 90 graus.

Uma outra dificuldade é o redimensionamento de escala da textura. Para uma imagem ser redimensionada nós consideramos que são efetuadas uma tupla de modificações



(a) Textura original

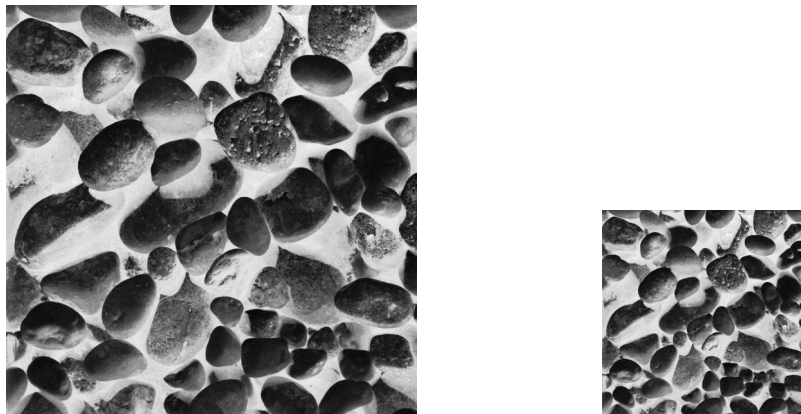
(b) Textura rotacionada em 90 graus

Figura 1 – Exemplo de uma aplicação de rotação em uma imagem de textura. Em (a) temos a imagem original sem rotação e em (b) o resultado da rotação quando a aplicamos com o valor de ângulo θ de 90 graus.

geométricas em relação a (X, Y) , tal que X seja a altura de uma imagem e Y seja a largura. Nas Equações (1.3) temos como entradas as posições originais (x, y) , os fatores de expansão (S_x, S_y) e, por fim, os novos valores (x', y') . Após modificação de escala, temos como saída as novas posições da imagem; quando os fatores (S_x, S_y) são maiores que 1, temos uma expansão; caso eles sejam inferiores a 1, teremos uma redução.

$$\begin{aligned} x' &= x.S_x, \\ y' &= y.S_y. \end{aligned} \tag{1.3}$$

As imagens podem ser expandidas ou reduzidas e para que aconteça uma transformação uniforme, os valores de (S_x, S_y) devem ser iguais. Na Figura 2 temos, à esquerda (a), a textura original, e à direita (b), a textura com suas dimensões reduzidas em 50%.



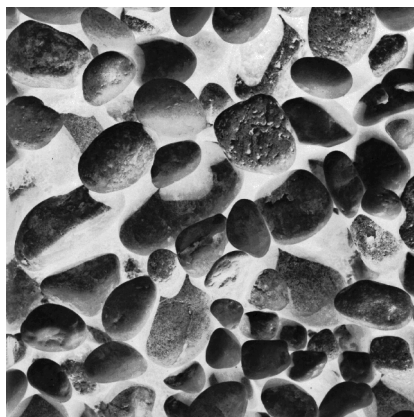
(a) Textura original

(b) Textura com dimensões reduzidas

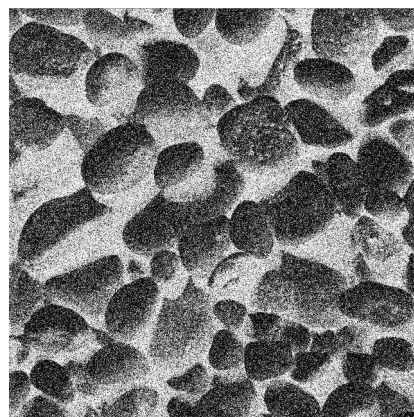
Figura 2 – Exemplo de redução da imagem da textura. Na Figura (a) temos a textura original, já na Figura (b) temos a mesma imagem com suas dimensões reduzidas em 50%.

Na Figura 2, temos um exemplo de redução uniforme, em que os valores de (S_x, S_y) foram iguais a 0,5, ou seja, ocorreu uma redução. Após a aplicação do processo de mudança de escala alguns problemas podem ocorrer, dos quais dois são importantes : (i) pixels que se sobrepõem por compartilharem as mesmas posições finais e (ii) posições na imagem final não são preenchidas por um pixel. Para contornar ambos os problemas existem diversas abordagens, sendo as mais comuns as que utilizam interpolação. Uma possível solução para uma imagem final com pixels não preenchidos por nenhuma intensidade é a aplicação de uma interpolação do vizinho mais próximo, ou seja, o pixel vazio assume o valor do pixel mais próximo que possua algum valor de intensidade. Quando pixels compartilham a mesma posição, em alguns casos uma simples média entre valores pode solucionar o problema. As posições finais podem não ser inteiras em decorrência dos fatores não serem inteiros também, uma forma de solucionar esse problema é o arredondamento no inteiro mais próximo para os valores de (x, y) . As soluções citadas não são fixas, haja vista existirem diversas abordagens que podem solucionar esses problemas, assim como também há outros métodos, além do geométrico, para mudança de escala (TUCERYAN; JAIN, 1993; SHAPIRO, 1989).

Por último temos os ruídos das imagens; um ruído em relação à textura pode ser considerado como um conjunto de variações aleatórias nas intensidades dos pixels que podem acabar deteriorando uma imagem e sua compreensão (ROHIT; ALI, 2013). Existem diversos tipos de ruídos, um dos mais simples de ser gerado é o aleatório, como pode ser visto na Figura 3, que simplesmente considera uma probabilidade fixa de cada pixel em assumir um valor aleatório e substituir seu valor original.



(a) Textura original



(b) Textura com ruído aleatório

Figura 3 – Exemplo de aplicação de ruído aleatório em uma imagem de textura. Na Figura (a) temos a imagem original sem ruído, na Figura (b) temos a imagem com ruído aleatório aplicado

Uma importante característica de um descritor é conseguir contornar todos os limitantes citados, sendo o mais robusto possível. Um descritor é considerado robusto se os limitantes citados interferirem minimamente em sua descrição, aumentando a probabili-

dade do sucesso na classificação.

1.2 OBJETIVOS

Este trabalho tem como objetivo geral analisar a criação de um descritor local eficiente para classificação de texturas, de tal forma que seja robusto quanto à rotação e à escala.

Temos os seguintes objetivos específicos:

- Analisar se as possíveis configurações (dimensão de vizinhança, total de vizinhos, etc) do descritor podem influenciar em sua robustez.
- Avaliar se o nosso método proposto é robusto à mudança de escala e rotação.
- Avaliar o desempenho em bases de texturas conhecidas para classificação.
- Determinar as melhores configurações para a aplicação eficiente do descritor proposto.

1.3 ESTRUTURA DO TRABALHO

Este trabalho é dividido em 5 capítulos, estruturados conforme a seguir:

- Capítulo 2: apresenta de forma mais detalhada a respeito de texturas, uma breve descrição da sua história, as formas mais usuais de serem descritas e trabalhadas. Apresenta um dos primeiros descritores relacionados à vizinhança, os descritores de Haralick e o GLCM. Apresenta o descritor local *Local Binary Pattern* (LBP). Nesse capítulo também é percorrido toda a história do LBP, desde a sua primeira versão até o segundo modelo, a inserção do invariante de rotação. Finalmente descreve de forma mais detalhada alguns métodos derivados do LBP.
- Capítulo 3: descreve como o método de origem *Local Vector Quantization Pattern* (LVQP) funciona, primeiro percorrendo os seus componentes vetor de diferença e quantização de vetores, para finalmente exemplificar como o método funciona em conjunto com tais componentes. Nesta parte do trabalho também é descrito o método proposto *Difference-Local Vector Quantization Pattern* (D-LVQP), como também os seus componentes, primeiramente relacionados ao novo conceito de vetor de diferença *Difference Quantization Pattern* (DQP) em conjunto o próprio LVQP e logo após é obtida a sua integração e funcionamento.
- Capítulo 4: descreve como o ambiente de teste foi criado, as variáveis que foram utilizadas, os métodos de teste e parâmetros. São listadas as bases em que foram executados os testes e descreve os resultados para cada uma delas com uma breve

análise. Nesse capítulo ainda é incluído uma comparação entre os métodos relacionados com aprendizagem profunda e o método proposto. Apresenta também uma validação estatística dos experimentos, utilizando os melhores resultados, e os que possuem maior diferença de acurácia do D-LVQP em relação ao LVQP.

- Capítulo 5: expõe as conclusões sobre o trabalho proposto e são discutidos os possíveis pontos de aperfeiçoamento.

2 ANÁLISE DE TEXTURAS

Neste capítulo são listados e analisados trabalhos da área relacionados a nossa proposta. As definições de textura são explicitadas com suas modificações e ambiguidades. Definimos os principais tipos de problemas relacionados à textura, assim como também listaremos o conjunto de definições da descrição de texturas que atualmente estão entre as mais aceitas. Nesta parte da dissertação será relatada a história de origem dos primeiros descritores visuais (Haralick e GLCM), e também do LBP em conjunto com seu precursor *Text Unit*. A primeira abordagem LBP é exemplificada com seus limitantes, modificações e funcionalidades. A segunda, e mais conhecida abordagem LBP, é explicada e comparada em relação às suas melhoras levando em consideração a primeira abordagem e seus avanços. A ferramenta de resistência à rotação do descritor LBP é descrita e exemplificada, concluindo, dessa maneira, uma listagem de trabalhos derivados ao LBP com o objetivo de cada uma e seus respectivos ganhos

2.1 DEFINIÇÃO DE TEXTURA

Em visão computacional e processamento de imagens, textura é um termo não completamente definido e existe uma antiga discussão nesse tema, sendo ela citada há mais de 50 anos por Julesz (JULESZ, 1962). O trabalho de Julesz foi um dos primeiros que apresentou uma definição de textura, mesmo que de forma vaga.

Julesz (JULESZ, 1962) argumentou que ela seria um conjunto de microrregiões e padrões. Somente após algum tempo, surgiram definições mais objetivas com relação ao que seria uma textura. Em 1973 Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973) estava desenvolvendo trabalhos com texturas relacionados a superfícies e relatou a perceptível importância das texturas para essa atividade e o desafio que era o seu processamento. Haralick (1973) afirmava que "textura é uma propriedade virtualmente nata de qualquer superfície - um pedaço de madeira, um pedaço de tecido, o padrão de cortes de um campo, etc". Apesar da textura ser de fácil descrição em termos empíricos para o homem, sua definição computacional é confusa e de difícil precisão (HARALICK; SHANMUGAM; DINSTEIN, 1973).

Como a textura é um termo de difícil definição, no decorrer dos anos surgiram diversas abordagens para descrição delas e de seus comportamentos. Coggis (1983) demonstrou essa confusão com uma coleção de definições de textura que podemos destacar.

- Podemos considerar textura como sendo um conjunto de microscópicas regiões. Essa estrutura é uma simples repetição de padrões em que cada elemento ou conjunto é arranjando de acordo com uma regra de posicionamento (TAMURA; MORI; YAMAWAKI, 1978).

- Uma região de uma imagem possui uma textura constante se um conjunto de estatísticas locais ou outras propriedades locais da imagem são constantes, com pouca variação ou aproximadamente periódicas (SKLANSKY, 1978).
- "Textura é aparentemente um notação paradoxal. Por um lado é comumente usada no início do processamento de informações visuais, especialmente para a prática da classificação. Por outro lado ninguém conseguiu produzir uma definição de textura comumente aceita. Para resolver esse paradoxo, achamos que dependeremos de um modelo mais sofisticado e rico, para o processamento de informações visuais prévias, que é um aspecto central no sistema que apresenta diversos níveis de abstração" (ZUCKER; KANT, 1981).
- A notação de textura depende de três componentes: (i) algum "padrão" que se repete por uma região grande em relação às dimensões da textura, (ii) os padrões são constituídos de arranjos não aleatórios de partes elementares, e (iii) os componentes são de forma geral entidades uniformes que possuem aproximadamente as mesmas dimensões (HAWKINS, 1969).

Com a falta de consenso na definição de textura, em 1990 a *Institute of Electrical and Electronic Engineers* (IEEE) lançou em seu guia de padrões uma definição. Segundo o IEEE (1990) textura seria um atributo que representa o arranjo espacial de níveis de cinza dos pixels em uma região quando relacionados a processamento de imagens. As diversas definições de textura dificultam o trabalho de análise matemática, sendo essa tarefa comumente realizada de forma empírica (MÄENPÄÄ, 2003). Mesmo com um definição não muito clara, é de comum acordo que a análise de textura é uma técnica de baixo nível, fundamental na área de Visão Computacional e reconhecimento de padrões (SAROJA; SULOCHANA, 2013). Tuceryan (1993) define que existem principalmente três problemas relacionados à textura: classificação, segmentação e síntese. Na Figura (4), podemos observar exemplos desses casos, à esquerda (a) temos um conjunto de cinco texturas distintas, no centro (b) as cinco texturas classificadas e, finalmente, à direita (c), o conjunto de texturas segmentadas.

O problema de classificação consiste da seleção de uma rótulo para um textura a partir um conjunto de rótulos determinados. A segmentação considera que a partir de um objeto de origem, a solução conseguirá determinar de acordo com alguns critérios os delimitantes de um conjunto de sub-regiões, que no nosso caso serão os tipos de texturas. O terceiro ponto abordado por Tuceryan (1993) se refere à síntese de texturas, uma área principalmente focada na compressão de dados e computação gráfica. A síntese de texturas pode ser relacionada à compressão, pois podemos recriar uma textura a partir de um conjunto de regras com poucas informações e para a computação gráfica com a possibilidade de recriação de texturas para renderização em superfícies.

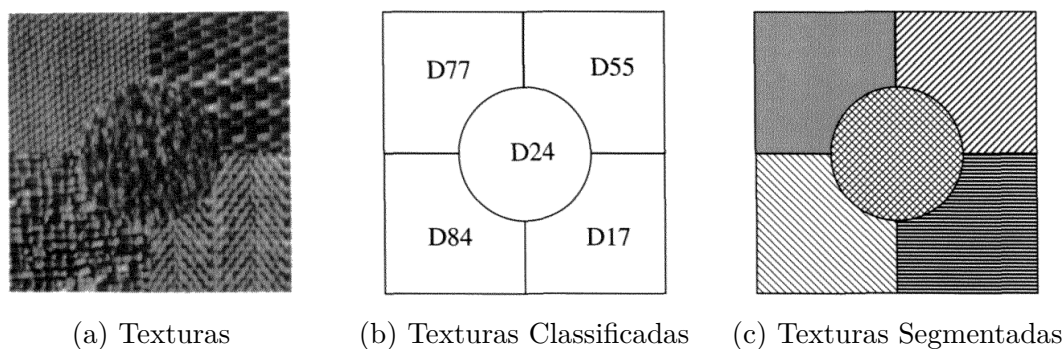


Figura 4 – Exemplo de aplicações com textura: Em (a) temos um conjunto de cinco texturas distintas. Uma aplicação possível seria a classificação em que cada textura receberia uma rótulo. Em (b). Um segundo caso possível seria a segmentação, na qual para cada tipo de textura seria definida uma região delimitante (c)

Fonte: (TUCERYAN; JAIN, 1993)

2.2 DESCRIÇÃO DE TEXTURAS

Uma abordagem comum para a resolução dos problemas com textura é extração de características. Ela seleciona informações relevantes segundo algum critério do conjunto de dados originais. Pode ser essencial, para que ocorra uma extração de características eficientes de uma textura, obter informação de como ela foi descrita. Semelhante à própria definição de textura, ocorrem algumas divergências quanto aos possíveis métodos de descrição de textura.

Gonzales (2006) afirma que existem três abordagens de descrição: estatística, estrutural e modelo. Xie (1998) define que existem cinco divisões: estatística, estrutural, espectral, modelo e filtros. Zhang (2010) define três tipos de métodos: estatístico, baseado em modelos e estruturais. Tuceryan (TUCERYAN; JAIN, 1993) define quatro divisões: estatístico, geométrico, baseado em modelos e métodos baseados em processamento de sinais. A enorme diversidade em definição de taxonomia dos métodos é um reflexo da falta de precisão na definição do que é textura e das diversas desconcordâncias na área. Dos métodos sugeridos três são os mais comuns: estatístico, estrutural e baseado em modelos, aos quais iremos definir nas seções a seguir.

2.2.1 Método Estatístico

Julesz (JULESZ, 1975) foi o precursor da abordagem estatística de texturas. Julesz (1975) afirma uma a relação de vizinhança de pixels com capacidade de reconhecimento de padrões. O método estatístico representa as texturas de forma não determinística com relação a alguma distribuição e também nas relações dos pixels vizinhos da imagem. Basicamente os métodos estatísticos podem ser divididos em três categorias (DIXIT; HEGDE, 2013):

- Primeira ordem: estatísticas e medidas extraídas de um pixel,

- Segunda ordem: estatísticas e medidas extraídas de dois pixels,
- Terceira ordem: estatísticas e medidas extraídas de três ou mais pixels.

Subsequentemente, podemos dividir o método estatístico em duas sub-categorias: na primeira temos uma análise individual dos valores dos pixels (variância, média, etc), na qual se inclui somente a primeira ordem; em segundo temos os métodos que levam em considerações as relações espaciais, como vizinhança e presença de padrões, onde estão presentes as abordagens de segunda ordem ou superiores. A maioria dos métodos atuais deriva das definições de Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973), levando em consideração correlações de pixels vizinhos e diferenças entre as suas intensidades. Dos diversos métodos estatísticos derivados, um dos mais famosos é o *Local Binary Pattern* (LBP), proposto por Ojala (OJALA; PIETIKAINEN; HARWOOD, 1994) em 1994 que teve como inspiração o método de Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973), modificado para uma abordagem com utilização de alta ordem, com microrregiões de vizinhança em conjunto com as diferenças de intensidade dos pixels.

2.2.2 Método Estrutural

O método estrutural define uma textura como um arranjo de "microtexturas", os denominados "*textons*" (HARALICK, 1979). As operações efetuadas nesse tipo de método são em sua maioria entre os próprios "*textons*", como adjacência, distância, periodicidade, etc. (MÄENPÄÄ, 2003), em relação aos seus níveis de cinza, homogeneidade, forma e/ou outras propriedades locais (LEE, 2004). Existem dois grandes problemas com esse tipo de abordagem (WECHSLER, 1980):

- Definir a micro textura primitiva que define a macrotextura
- Definir que regra de repetição que possibilite uma replicação

O primeiro problema é facilmente resolvido se a textura for simples ou sintetizada, ou seja, a textura possui seus agrupamentos bem definidos com poucas variações. O segundo problema demanda um pouco mais de análise; é necessário avaliar se é possível replicar a textura baseado na micro região primitiva, e qual tipo de distribuição seria a mais adequada para essa tarefa (WECHSLER, 1980). Caso os limitantes citados não estejam presentes, a abordagem estrutural pode apresentar bons frutos, podendo possibilitar uma boa descrição simbólica dos padrões a serem descritos com uma certa facilidade. Como o método estrutural apresenta limitações, é esperado que ele se comporte de forma melhor quando utilizado em problemas de síntese de texturas, ou em situações que envolvam texturas naturais. (MATERKA; STRZELECKI, 1998).

2.2.3 Método Baseado em Modelo

Métodos baseados em modelo são focados não somente na construção dos modelos matemáticos que possam descrever a textura, mas também na sua síntese (XIE; MIRMEHDI, 1998). Para a geração dos modelos podem ser utilizadas abordagens fractais e estocásticas para interpretação das texturas (MATERKA; STRZELECKI, 1998). Os problemas relacionados à geração de modelo estão diretamente ligados com a definição dos parâmetros que melhor se aproximem do modelo ideal capaz de descrever a textura (XIE; MIRMEHDI, 1998). Dentre os métodos que se incluem nesse tipo de modelo podemos citar além dos fractais, modelos autorregressivos, modelos de campos aleatórios e modelos em miniatura (MATERKA; STRZELECKI, 1998).

Dentre os métodos existentes dois são de bastante importância: o método fractal e os campos aleatórios de Markov. O método fractal foi proposto em 1983 por Mandelbrot (MANDELBROT, 1983), ele observou a presença desse tipo de modelo na natureza, que mesmo com mudança de escalas mantinha as mesmas propriedades. Um ponto relevante em que a abordagem de Mandelbrot 1983 perde desempenho é na capacidade de representação de estruturas locais em texturas não naturais, já que considera as texturas ambientes como mudanças de escala.

O segundo modelo é o baseado em campos aleatórios de Markov ou *Markov Random Fields* (MRFs). Nesse tipo de abordagem, é julgado se as informações locais podem definir de forma satisfatória uma representação global da textura (MATERKA; STRZELECKI, 1998). O MRF apresenta a informação da probabilidade *a priori* de dependências entre padrões, como os presentes em textura ou características de objetos. Um modelo MRF favorece sua própria classe de padrões associados a uma alta probabilidade em relação a outras classes existentes. Métodos baseados em MRF utilizam abordagens estatísticas para estimação de funções objetivas para decisão dos ótimos dos modelos e posterior geração dos padrões (LI, 2009).

Como podemos observar neste capítulo, mesmo que a primeira definição de textura seja antiga, esse tópico continua se mantendo aberto com espaço para discordância entre autores. De forma paralela com o conjunto de definições de textura, surgiram diversos métodos para descrever uma textura, ou seja, como uma textura pode ser formada e quais são os seus componentes. Apesar das divergências na áreas, nós podemos afirmar que existem similaridades indiscutíveis, sendo uma das principais a que afirma ser a textura uma característica de baixo nível e que a classificação, segmentação e síntese de texturas pode ajudar a resolver uma gama de problemas.

2.3 DESCRITORES DE HARALICK

Mesmo sem possuir uma definição clara de textura, vários pesquisadores tentaram trabalhar com os conceitos relacionados a ela. Um dos mais importantes estudos foi desenvolvido

por Haralick e por seus grupos de pesquisa ainda nos anos 70. Publicado no ano de 1973, (HARALICK; SHANMUGAM; DINSTEIN, 1973), no trabalho foi considerado a probabilidade marginal da matriz de co-ocorrência, sendo essa a forma de medir as características da texturas. Essa maneira de utilização é o que chamamos de média, que foi utilizada para a criação do que chamamos descritor textural.

Segundo Haralick (1973), é possível obter um total de 14 descritores texturais a partir da matriz de co-ocorrência:

- Segundo Momento Angular
- Contraste
- Correlação
- Variância
- Homogeneidade
- Soma das Médias
- Soma das Variâncias
- Soma das Entropias
- Entropia
- Variância da Diferença
- Entropia da Diferença
- Energia
- Probabilidade Máxima
- Diferenças de Ordem K

Esses extratores são obtidos a partir de uma distribuição espacial dos pixels da textura com a geração da matriz de co-ocorrência. Para a geração da matriz de co-ocorrência são utilizados os elementos que se relacionam com a frequência da presença de um determinado tom de cinza, a , na textura com outro tom, b , separados por uma distância d_x de linhas e d_y de colunas. Essa medidas consideram que ocorrem repetição de padrões na textura, considerando que existe uma relação entre os blocos de pixels e vizinhanças. Na Figura (5) temos um exemplo de como os extratores podem ser computados desde da imagem original, mais à esquerda, da criação da matriz de correlação, da normalização até a computação dos descritores, à direita.

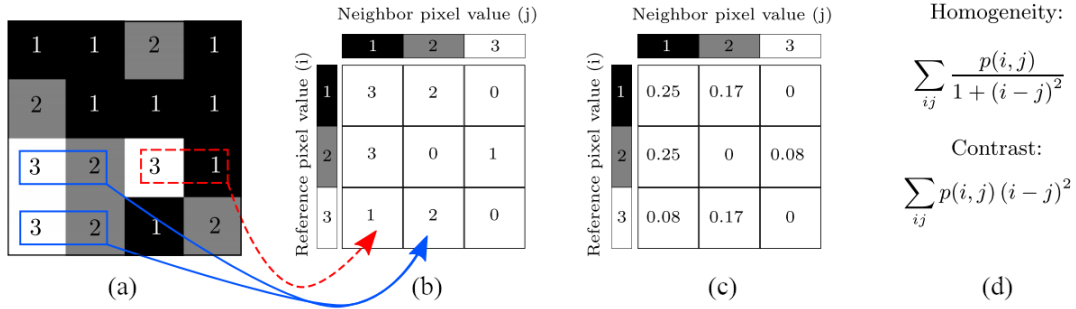


Figura 5 – Exemplo da aplicação dos descritores de Haralick. Primeiro temos a textura original (a), depois geramos a matriz de correlação entre os pixels e seus vizinhos, no caso (b) é computada a relação com o vizinho a direita foi o considerado, depois a matriz é normalizada entre 0 e 1 (c) e como passo final são computados os descritores (d).

Fonte: (BRYNOLFSSON *et al.*, 2017)

Como visto na Figura (5) primeiro uma textura é selecionada, depois a relação com seus vizinho é computada. Esse método é uma abordagem de segunda ordem, ou seja, somente uma vizinho é considerado, mas quatro matrizes são criadas (horizontal, vertical, diagonal à esquerda e diagonal à direita). Depois de computado as matrizes, os descritores são extraídos, para serem utilizados como características em algum problema relacionado a textura.

Os descritores de Haralick são uma ótima ferramenta que vem sendo utilizada há muito tempo, porém alguns dos seus descritores podem demandar um grande custo computacional e podem acabar não agregando uma descrição relevante que justifique a sua utilização. O *Level Co-occurrence Matrix* (GLCM) efetua algo semelhante a essa seleção, somente alguns descritores são considerados, assim como também é possível escolher quais direções irão compor as matrizes de correlação. Na Figura (6) temos um exemplo de como o GLCM funciona, desde a textura original até a extração dos descritores e matrizes.

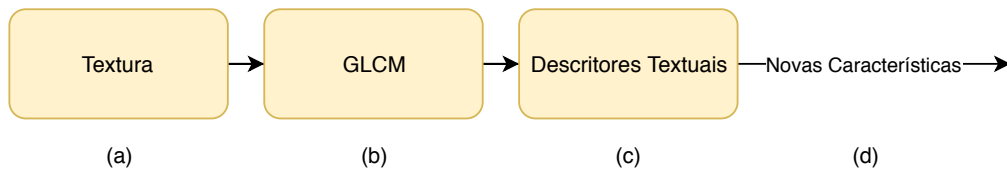


Figura 6 – Exemplo da aplicação dos GLCM. Primeiro temos a textura original (a), depois selecionamos os descritor utilizados no GLCM (b), geramos as matrizes de correlação e efetuamos a extração dos descritores (c) e finalmente obtemos as novas caraterísticas da textura (d).

O GLCM consegue representar de uma forma aceitável, mas apesar de ser de fácil implementação por vezes não se torna uma saída viável, dado que pode demandar um grande custo computacional. Outros descritores podem apresentar desempenho igual ou superior com muito menos esforço. Um desses descritores é o *Local Binary Pattern* (LBP)

que se baseia na ideia de vizinhança como o GLCM de simples implementação, mas de baixo custo computacional.

2.4 O LOCAL BINARY PATTERN - LBP

O *Local Binary Pattern* (LBP) é um descritor local, que pode ser visto como a união da abordagem dos modelos estatístico e estrutural que são tradicionalmente divergentes (PIETIKÄINEN; ZHAO, 2016). O LBP é uma abordagem amplamente aplicada por possuir robustez às mudanças de iluminação, à baixa complexidade, e à capacidade de descrever em diversos casos pequenas modificações ou detalhes (NANNI; LUMINI; BRAHNAM, 2012). O LBP foi proposto inicialmente em 1994 (OJALA; PIETIKAINEN; HARWOOD, 1994); nesse trabalho, ele foi descrito de uma forma mais simples que a atual, mas já apresentava uma considerável invariância à mudança de iluminação e à mudança de escala. O LBP possui uma grande influência da definição de *Text Unit* proposto no início dos anos 90 por He e Wang (HE; WANG, 1990).

A *Text Unit* é uma abordagem estrutural que considera que uma textura é formada por um conjunto de microestruturas denominadas *Text Unit* (TU). As TU são pequenas unidades que conseguem extrair informações locais da textura e informações em relações aos seus vizinhos. Com a TU as informações das texturas são extraídas de uma submatriz com dimensão 3x3 como pode ser visto na Figura 7, com um conjunto de 9 valores $V = \{V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$, onde V_0 é o valor de intensidade do pixel central e o restante dos valores são os níveis de intensidade dos pixels que o circundam.

V_1	V_2	V_3
V_8	V_0	V_4
V_7	V_6	V_5

Figura 7 – Exemplo da estrutura de um TU 3x3. Cada célula representa a intensidade de um pixel, sendo o V_0 o pixel central, e todos as outras células que o circundam, seus vizinhos, com seus respectivos valores de intensidades.

Como podemos observar a partir da Figura 7 conseguimos extrair o vetor de valores V . O vetor de características $TU = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8\}$ é extraído a partir do vetor V de intensidades dos pixels sendo E_i definido segundo a Equação (2.1).

$$E_i = \begin{cases} 0 & \text{se } V_i < V_0, \\ 1 & \text{se } V_i = V_0, \\ 2 & \text{se } V_i > V_0. \end{cases} \quad (2.1)$$

Na Equação (2.1), V_i é o valor de intensidade de um vizinho e V_0 é o valor do pixel central, na equação é possível obter 3 valores $E_i = (1, 2 \text{ ou } 3)$. Podemos observar a

aplicação do método na Figura 8 da codificação TU, à esquerda (a) temos um bloco de pixels de entrada com os valores $V = \{70, 112, 230, 64, 70, 152, 64, 104, 112\}$, à direita (b) temos o resultado de saída, ou seja, o vetor de características $TU = \{2, 2, 0, 1, 2, 0, 2, 0\}$.

112	230	64
59	70	70
104	64	152

(a) Bloco de Pixels

2	2	0
0		1
2	0	2

(b) Codificação TU

Figura 8 – Exemplo aplicações da codificação TU. Em (8a) temos o bloco de pixels de exemplo com o valor do pixel central $V_0 = 70$, em (8b) temos o bloco de pixels processados com os novos valores componentes do vetor de características E

Como pode ser observado, os componentes da TU podem assumir 3 valores (0, 1 ou 2), ou seja, temos o total de 3^8 possibilidades da TU, resultando em um total 6561 tipos de TU. Para a rotulação cada TU recebeu um valor único que é denominado *Unit Number*, segundo a Equação (2.2). A equação tem como entrada o vetor de características TU e possui como saída um valor escalar como rótulo para o vetor.

$$N_{TU}(E) = \sum_{i=1}^8 E_i(3^{i-1}). \quad (2.2)$$

Considerando $TU = \{E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8\}$ na Equação (2.2), TU é o vetor de entrada com E_i sendo cada um dos componentes do vetor. Com a extração dos *Unit Numbers* é efetuado a extração de espectro, que consiste basicamente na criação de uma matriz de todos os *Unit Numbers* e na avaliação com que frequência cada padrão se apresenta, de forma geral essa avaliação é efetuada com a criação de um histograma. A Figura 9 exemplifica como o processo ocorre para uma textura, desde o início da extração com codificação, rotulação até a criação do histograma.

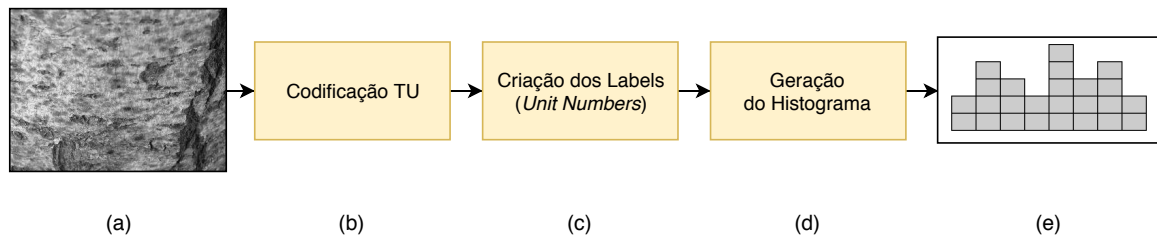


Figura 9 – Fluxograma da abordagem TU. Primeiro a partir da textura (a) aplicamos a codificação TU (b); em seguida, para cada código aplicamos a codificação *Unit Number* (c), logo após extraímos o histograma da textura (d), e, por final, obtemos o histograma como vetor final de característica. (e)

2.5 PRIMEIRA ABORDAGEM LBP

Tomando como base o *Texture Unit*, Ojala propôs a primeira abordagem do seu método *Local Binary Pattern* (LBP) em 1994 (OJALA; PIETIKAINEN; HARWOOD, 1994), que descrevia a relação entre o pixels e sua vizinhança. A ideia principal da primeira abordagem do LBP recai sobre o conceito de que texturas podem ser definidas como duas medidas complementares, uma sendo o micro padrão e a outra sendo o contraste. Uma importante característica adquirida foi a uma menor interferência na descrição quando referente à variação brusca dos tons de cinzas, ocasionada pela mudança de iluminação.

Similar ao TU, a primeira versão do LBP trabalha com um bloco de pixels de dimensão 3x3, utilizando o pixel central como limiar do bloco. Um dos diferenciais da abordagem TU, era função de codificação que deixa de possuir uma saída ternária (1,2 ou 3) e passa a ser uma função de limiarização com saída binária (1 ou 2). Podemos observar na Equação (2.3) a função de limiarização, tendo como entrada o valor de um pixel vizinho e tendo como saída um escalar referente à limiarização do pixel central.

$$E_i = \begin{cases} 1 & \text{se } V_i \geq V_0, \\ 0 & \text{se } V_i < V_0. \end{cases} \quad (2.3)$$

Na Equação (2.3), temos que V_i é um vizinho arbitrário, V_0 é o pixel central e tem como saída um valor binário (0 ou 1). Com a limiarização efetuada, os vizinhos codificados são multiplicados por uma matriz de pesos já definida pelo trabalho e, finalmente, é efetuado a sua soma. Na Figura 10 temos a exemplificação do processo. À esquerda (a), está a textura original, logo em sequência (b) a limiarização, e posteriormente, o bloco de pesos (c) e, como produto, o resultado: a soma da multiplicação dos valores limiarizados pelos pesos (d).

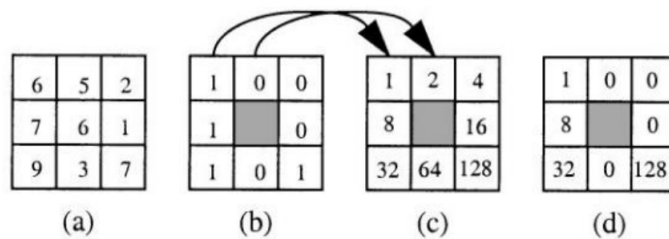


Figura 10 – Exemplo da codificação LBP original. Primeiro, temos o bloco de textura original (a); em seguida, aplicamos a limiarização nos vizinhos (b) e multiplicamos pelos pesos já definidos (c) e obtemos os novos valores da codificação (d).

Fonte: (OJALA; PIETIKÄINEN, 1999)

Com os valores obtidos da codificação, obtemos o número LBP, ou seja, a soma dos resultados da multiplicação dos pesos pelos valores limiarizados. Podemos definir o processo final de criação do código LBP na Equação (2.4), que tem como entrada um conjunto de

vizinhos limiarizados e o conjunto de pesos definidos e possuindo como saída o código LBP.

$$LBP_n(E, P) = \sum_{i=1}^8 E_i P_i. \quad (2.4)$$

Na Equação (2.4), temos como entrada um vetor limiarizado de vizinhos com valores binários (0 ou 1), e uma matriz de pesos já definida $P = \{1, 2, 4, 16, 128, 64, 32, 8\}$, onde E_i é i-ésimo valor do vetor de pixel limiarizados e P_i é o i-ésimo valor do vetor de pesos. Para o exemplo da Figura 10, temos que o vetor de vizinhos [e $E = \{1, 0, 0, 0, 1, 0, 1, 1\}$] com o vetor de pesos já definido $P = \{1, 2, 4, 16, 128, 64, 32, 8\}$, aplicando a função presente na Equação (2.4) temos que $LBP_n(E, P) = (1 * 1) + (0 * 2) + (0 * 4) + (0 * 16) + (1 * 128) + (0 * 64) + (1 * 32) + (1 * 8) = 1 + 128 + 32 + 8 = 169$, sendo agora 169 o novo valor do pixel central em uma nova matriz. Apesar do LBP original possuir um poder de descrição alto em relações aos métodos da época, ele ainda possuía uma fraca robustez à rotação por seu método ser dependente de um conjunto de pesos fixos.

2.6 LBP - NOVO MODELO

A abordagem original do LBP possui alguns limitantes, sendo um dos principais, a geração de descrição similar de padrões diferentes quando rotacionados. Essa limitação foi contornada pelo próprio Ojala *et al* em 2002 (OJALA; PIETIKAINEN; MAENPAA, 2002), que modificou a estrutura inicial do LBP, criando o LBP invariante de rotação uniforme. A nova abordagem invariante não somente acrescentou robustez a rotação como também a ruídos. De forma similar ao LBP original, a nova abordagem possui como saída um valor inteiro para cada bloco de pixels de uma imagem. O novo vetor de características dessa imagem é um histograma dessa imagem codificada. O histograma gerado possui um total de 256 níveis, sendo esse os valores dos novos pixels, indo do 0 ao 255. O processo de codificação LBP pode ser observado na Figura 11, na qual primeiro nós obtemos a textura original à esquerda (a), depois extraímos a codificação LBP ao centro (b), e por fim computamos o histograma da imagem codificada à direita (c).

Para a compreensão da nova abordagem LBP, dois conceitos são necessários: raio e vizinhos. Vizinhos são todos os pixels que estão a uma distância de raio fixa do pixel central. Para o LBP, P vizinhos são selecionados a partir de uma distância de raio R. Na Figura 12 nós temos exemplos da aplicação do LBP, onde cada bloco de pixel representa uma configuração diferente dos valores do raio e da dimensão da vizinhança.

As posições dos pixels podem se interpretadas como posições de uma matriz. Cada pixel obtém um valor de coordenada da matriz referente a sua coluna e linha. Como os vizinhos circundam o pixel central de forma circular suas posições são determinadas pela equação da circunferência. A equação da circunferência pode ser observada na Equação

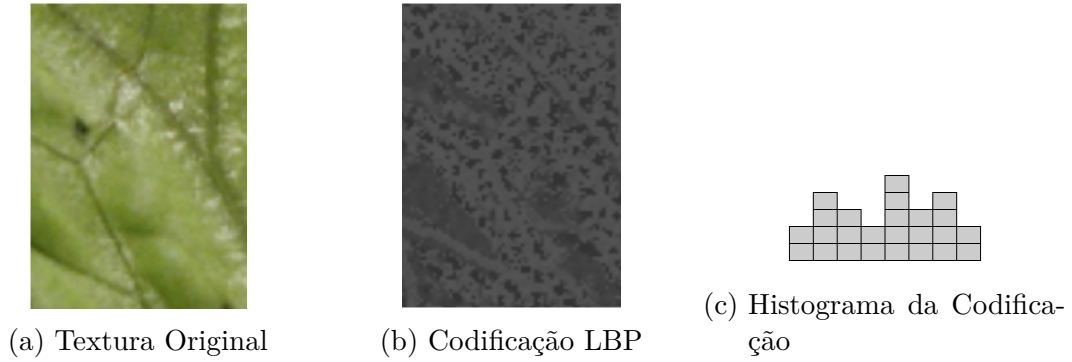


Figura 11 – Exemplo da aplicação da codificação LBP. Primeiro temos como entrada uma textura original (a) sem codificação; em seguida, aplicamos a codificação LBP na textura (b) e o histograma da textura codificada é calculado (c).

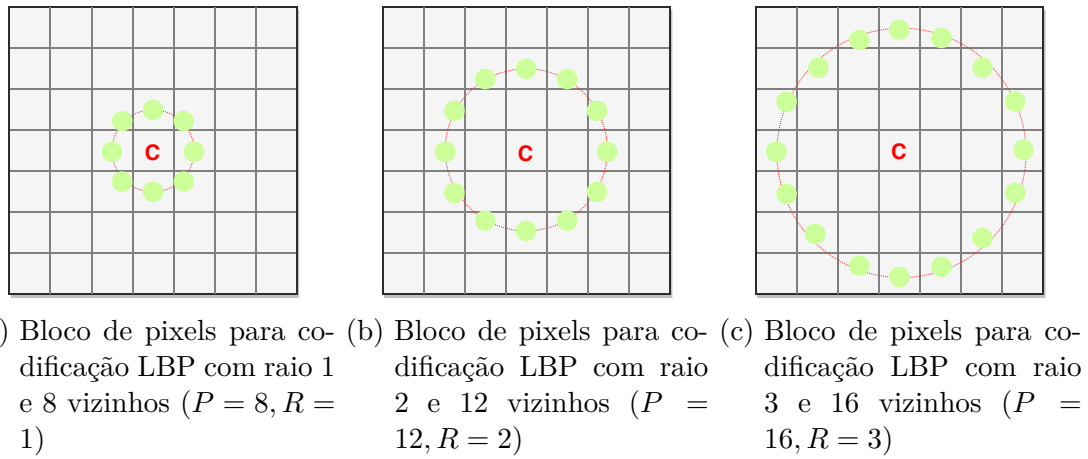


Figura 12 – Exemplo de abordagem LBP com variação de raio e total de vizinhos.

(2.5), que tem como entrada o valor do raio, o total de vizinhos e posição do vizinho no vetor de vizinhos e tem como saída as posições na matriz do vizinhos selecionados.

$$g_p(R, P, p) = (R \sin(2\pi p/P), R \cos(2\pi p/P)). \quad (2.5)$$

Na Equação (2.5), temos que R é o raio, P é o número total de vizinhos e p é o índice do vizinho selecionado, tal que $p \in \{0, 1, 2, \dots, P - 1\}$. A função retorna à posição (x, y) do pixel p na matriz. Um problema surge quando as posições resultantes da equação não são inteiras, sendo necessário a utilização de algum método para gerar o novo valor desse pixel. Neste trabalho foi utilizada a interpolação bilinear. A Figura 13 demonstra uma posição (x, y) não inteira entre os pixels nas posições (x_1, y_1) e (x_2, y_2) , na qual a interpolação retorna o novo valor do pixel baseado nessas posições.

Na Figura 13, p é o valor de intensidade do pixel na posição (x, y) que foi determinado pela equação da circunferência. Após aplicação da interpolação bilinear, o seu resultado é a nova intensidade do pixel na posição (x, y) . No conjunto de Equações 2.6 podemos observar como é aplicada a interpolação bilinear, tendo como entrada as posições definidas pela equação da circunferência (x, y) e como saída uma nova intensidade para o pixel

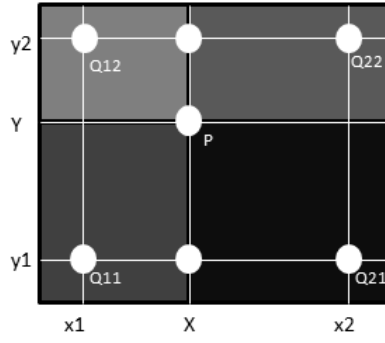


Figura 13 – Exemplo da aplicação da interpolação bilinear, considerando os vizinhos próximos nas posições (x_1, y_1) , (x_1, y_2) , (x_2, y_1) , (x_2, y_2) , seus valores de intensidade respectivamente $(Q_{11}, Q_{12}, Q_{21}, Q_{22})$ e, por fim, o novo valor de pixel p nas posições (x, y) definidos pela interpolação bilinear.

central.

$$f(x, y) \cong \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2), \quad (2.6a)$$

$$f(x, y_1) \cong \frac{x_2 - x}{x_2 - x_1} Q_{11} + \frac{x - x_1}{x_2 - x_1} Q_{21}, \quad (2.6b)$$

$$f(x, y_2) \cong \frac{x_2 - x}{x_2 - x_1} Q_{12} + \frac{x - x_1}{x_2 - x_1} Q_{22}. \quad (2.6c)$$

No conjunto de Equações (2.6), Q_{11} , Q_{12} , Q_{21} e Q_{22} são intensidades de pixels próximos à posição de entrada (x, y) e considerados os vizinhos próximos (x_1, y_1) , (x_1, y_2) , (x_2, y_1) , (x_2, y_2) com posições inteiras mais próximos. A interpolação bilinear é uma derivada em relação a (x, y) . O novo valor do pixel é definido por uma relação de distância dos pixels vizinhos com o pixel na posição resultante da equação de circunferência. Quanto mais próximo o pixel vizinho for do pixel selecionado da equação de circunferência, maior é a sua influência no novo valor de intensidade do pixel.

Com os conceitos bem definidos do que são vizinhos e raio e como os valores dos vizinhos são obtidos a partir de interpolação, podemos descrever o novo LBP. A codificação LBP percorre a imagem, extraindo a cada passo um bloco de pixels com raio R e P vizinhos. A nova equação da codificação LBP abre mão da matriz de pesos da sua versão anterior e aplica uma conversão binária aos números resultantes da limiarização. A nova equação de codificação pode ser vista na Equação (2.7), onde se obtém as diferenças dos vizinhos em relação ao pixel central e aplica a limiarização. A entrada da equação é o valor do pixel central c e tem como saída um valor inteiro convertido a partir da limiarização das diferenças com a Equação (2.8) para a base 10.

$$LBP_{P,R}(c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p. \quad (2.7)$$

Na Equação (2.7), P é o total de vizinhos, R o raio, g_c é o valor da intensidade do pixel central e g_p representa a intensidade do p -ésimo vizinho. O resultado da soma é o novo código LBP. A função de binarização do pixel central $s(x)$ é definida na Equação (2.8). O valor de x é o valor da diferença do pixel central em relação a algum vizinho.

$$s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (2.8)$$

Na Equação (2.8) o valor resultante é 0 se a diferenças entres os vizinhos, x , for menor que 0 ou em caso contrário é 1. A codificação LBP é definida de forma completa na Figura 14. Onde primeiro selecionamos uma bloco de pixels (a), então extraímos o valores (b) da binarização (c), e por final convertemos o valor para a base 10 (d).

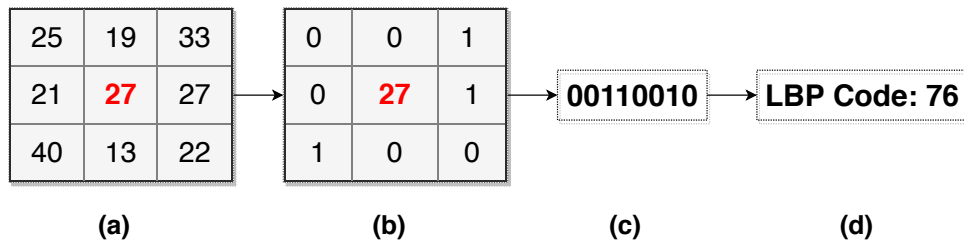


Figura 14 – Exemplo do processo da geração do código LBP a partir de um bloco de pixels e de um novo valor do pixel central. Primeiro temos o bloco de pixel original (a) então obtemos os valores da binarização dos vizinhos (b) a partir da binarização obtemos uma vetor binários (c) que como saída é convertido para base 10 se tornando o novo código LBP (d)

Para o exemplo da Figura 14, primeiro obtemos o valor do pixel central, que no caso será 27, então aplicamos a Equação (2.7) com o valor de raio $R = 1$ e o valor de vizinhos $P = 8$. Após a aplicação da Equação (2.8) de limiarização, obtemos um vetor binário (00110010), que no fim é convertido em um inteiro pela soma presente na Equação (2.7), tendo com saída resultante $0 \times 1 + 0 \times 2 + 1 \times 4 + 1 \times 8 + 0 \times 16 + 0 \times 32 + 1 \times 64 + 0 \times 128 = 76$.

Similar ao LBP original (OJALA; PIETIKAINEN; HARWOOD, 1994), para extração de características foram gerados histogramas das imagens codificadas. A geração do histograma é fácil e rápida de ser adquirida, além de se provar uma ótima ferramenta para problemas de classificação e descrição. Como produto final da codificação LBP, podemos ver, na Figura 14 o resultado com os pixels codificados para a extração das novas características. Uma visão completa da codificação LBP pode ser observada na Figura 15.

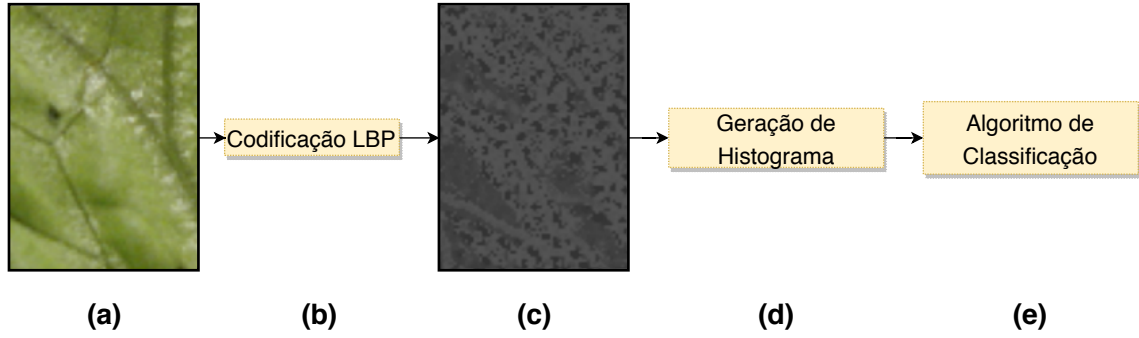


Figura 15 – Fluxograma geral do LBP. Primeiro selecionamos a textura original (a), então a imagem é codificada pelo LBP (b) que tem como saída a imagem codificada (c); da imagem codificada, é então extraído o histograma (d) que é utilizado como conjunto de característica para algum algoritmo de classificação(e).

2.7 INVARIANTE DE ROTAÇÃO UNIFORME

Uma forma de contornar a pouca robustez para a rotação do LBP foi proposta no mesmo artigo de Ojala em 2002 (OJALA; PIETIKAINEN; MAENPAA, 2002). Para compreender como o problema foi abordado, é necessário primeiro entender que a rotação de um padrão é simplesmente um *bitwise*, ou seja, uma sequência de trocas dos valores do vetor binário entre 0 e 1. Cada mudança de valor de um componente do vetor binário após uma rotação é o que denominamos de transição espacial. Se, após uma rotação no sentido anti-horário do vetor binário de origem, for obtido como resultante o vetor rotacionado com um total de transições espaciais menor ou igual a α esse padrão será uniforme em relação a α . Ojala (2002) alega que existem propriedades fundamentais e essenciais em padrões uniformes. Se um padrão não é uniforme para o valor de α , é mais provável que dentre os padrões locais ocorra alguma transição espacial. Quando o número de transições espaciais se torna elevado em um padrão, com o total transições espaciais maior que α , esse padrão acaba por dificultar a tarefa da classificação, já que as possíveis rotações acabam se tornando padrões diferentes.

Se consideramos um exemplo de código LBP igual a 10010100, nesse caso esse código é o mesmo padrão que o código 00101001, mas com uma rotação em sentido anti-horário. Para efetuar a rotação no código anterior é necessário a mudança do valor de 6 bits. Podemos observar na Equação (2.9) a fórmula para calcular o total de transições espaciais presentes em um código. Na Equação (2.9), temos como entrada o código binário resultante da função do LBP, com os valores de raio e vizinhos definidos.

$$\begin{aligned}
 U(LBP_{P,R}(c)) = & |s(g_{P-1} - g_c) - s(g_0 - g_c)| \\
 & + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|.
 \end{aligned} \tag{2.9}$$

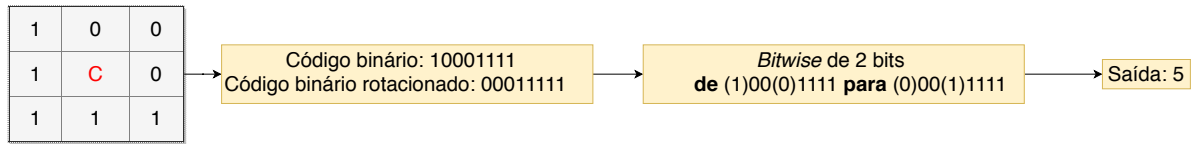
Na Equação (2.9), P é o total de vizinhos, R é a dimensão do raio, g_p o pixel a ser computado, g_c o pixel central. A função $s(x)$ é simplesmente a função de binarização que

pode ser vista na Equação (2.8).

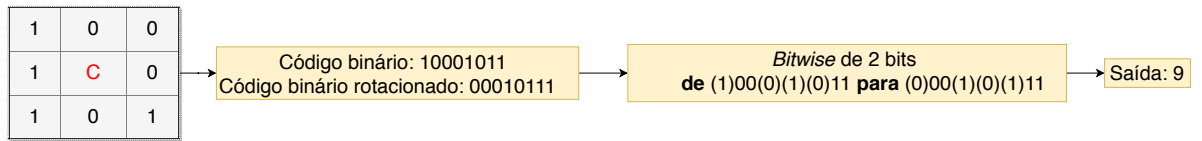
Complementando a nova proposta, Ojala (2002) considera que um padrão é uniforme se o total de transições espaciais for inferior ou igual a α , retornando o valor do somatório do código binário. Caso o padrão não seja uniforme, então a ele é atribuído o valor de $P + 1$. Na Equação (2.10) podemos observar como o processo ocorre, com a codificação LBP uniforme.

$$LBP_{P,R}^{riu\alpha}(c) = \begin{cases} \sum_{p=1}^P s(g_p - g_c) & \text{se } U(LBP_{P,R}(c)) \leq \alpha, \\ P + 1 & \text{senão.} \end{cases} \quad (2.10)$$

Na Equação (2.10), P é o total de vizinhos, R é a dimensão do raio, g_p o pixel a ser computado, g_c o pixel central. A função $s(x)$ é a função de binarização, como pode ser vista na Equação (2.8). Se o total de transições espaciais for menor ou igual ao valor de α , então a saída é o valor da soma do vetor binário segundo a função de binarização $s(x)$, caso contrário se tem como retorno o valor de $P + 1$. Na Figura 16, nós podemos observar um exemplo para padrões uniformes com $\alpha = 2$: primeiro temos o padrão (10001111), que é uniforme e retorna a soma dos valores que o compõem $1 + 0 + 0 + 0 + 1 + 1 + 1 + 1 = 5$; no segundo exemplo, temos um padrão que não é uniforme, retornando o valor de $P + 1 = 8 + 1 = 9$



(a) Exemplo de código com 2 transições espaciais, logo um código uniforme para ($\alpha = 2$), tendo com saída o código 5.



(b) Exemplo de código com 4 transições espaciais, logo um código não uniforme para ($\alpha = 2$), tendo com saída o código 9.

Figura 16 – Exemplo da aplicação do invariante de rotação com transições espaciais. No exemplo temos dois códigos diferentes, em (a) temos um código uniforme, ou seja, é uniforme, retornando a soma do vetor binário igual a 5, no segundo caso (b) temos 4 transições espaciais, ou seja, não é uniforme, retornando o valor de $P + 1 = 8 + 1 = 9$.

2.8 VARIANTES DO LBP

Similar a Ojala *et al.* (OJALA; PIETIKAINEN; HARWOOD, 1994; OJALA; PIETIKAINEN; MAENPAA, 2002), diversos trabalhos foram produzidos baseados na definições do LBP para

o melhoramento de diversos pontos. Podemos citar o trabalho de Tan e Triggs, denominado *Local Ternary Pattern* (LTP), que é basicamente um extensão do LBP que possui uma codificação binária (0 ou 1) para uma codificação ternária (-1, 0 ou 1); esse trabalho melhorou a resistência ao ruído e manteve a simplicidade do LBP (TAN; TRIGGS, 2010). O método *Extended Local Binary Pattern* (ELBP) tem como característica a presença de mais informações sobre os micro padrões; ou seja, o método leva em consideração outros padrões vizinhos em relação ao pixel central (LIU *et al.*, 2012). O ELBP apresentou uma desempenho superior em relação ao LBP original, com um aumento de robustez ao ruído, além de demonstrar que a combinação de características complementares pode trazer bons frutos. O *Complete Local Binary Pattern* (CLBP) é um método em que cada componente do código LBP é considerado com uma nova característica individual, sendo elas: magnitude, sinal e valor do pixel central. Similar ao ELBP o CLBP conseguiu elevar a robustez ao ruído e ajudou no crescimento da generalização das características (GUO; ZHANG; ZHANG, 2010).

O método denominado como *Dominant LBP* (DLBP) considera que padrões que aparecem com menor frequência devem ser descartados para a criação do histograma. A abordagem DLBP conseguiu uma acurácia superior em relação ao LBP original, acréscimo de robustez a ruído, além de reduzir a quantidade de dados para o processamento (LIAO; LAW; CHUNG, 2009). O trabalho de Qi *et al.* consite em uma análise em grande escala dos componentes estruturais da textura, em um maior contexto espacial (vizinhos próximos, limiares para binarização, etc), aumentando a acurácia e robustez a ruído (QI *et al.*, 2014).

Utilizando quantização de vetores para classificação de texturas temos o trabalho de Hussai e Triggs (HUSSAIN; TRIGGS, 2012), que se baseia na utilização de técnicas de clusterização para redução de dimensionalidade e criação de novas características, conseguindo um aumento de acurácia além de um redução do tempo de processamento. Ainda no mesmo ramo de quantização de vetores temos o *Local Vector Quantization Pattern* (LVQP) (PAN; FAN; ZHANG, 2015), que utiliza a quantização de vetores para redução de características, aumento de robustez ao ruído e rotação. A abordagem original do LBP é para imagens em escala de cinza, porém Lee *et al.* propôs uma abordagem de uma variante do LBP para imagens coloridas que foi aplicado no reconhecimento de faces (LEE *et al.*, 2012). O método de Lee *et al.* apresenta um aumento de acurácia para o reconhecimento de faces em relação a outros métodos baseados no LBP. Um dos trabalhos mais atuais derivados do LBP foca em uma modificação da conhecida abordagem de 2 dimensões para uma com 3 dimensões, sendo essa terceira dimensão criada por conjunto de codificações LBP com algoritmos gulosos para o crescimento (CITRARO *et al.*, 2017)CITRARO20178, melhorando a capacidade de reconhecimento de texturas e robustez a ruídos.

Como podemos observar neste capítulo, o LBP é um poderoso e simples descritor local,

que pode ser aplicado nas mais diversas atividades. Nessa parte do trabalho foi explicitado o desenvolvimento do LBP desde a sua primeira versão com dependência de pesos, a segunda utilizando o pixel central com o limiar, até a sua última versão com a abordagem para o problema da rotação de padrões. Foram listados diversos trabalhos relevantes que tiveram o LBP como origem, além de apresentar as diversas técnicas aplicadas em cada uma delas. Obtivemos um vislumbre da importância do LBP e como ele modificou a compreensão de texturas e atividades relacionadas a estas.

3 MÉTODO PROPOSTO

Este capítulo apresenta a origem do LVQP e seus componentes, a definição do conceito de vetor de diferença, um importante componente do LVQP, além de apresentar também a quantização de vetores e sua relação com outros métodos de clusterização. Demonstramos como todos os componentes trabalham de forma conjunta para o funcionamento do LVQP.

Nesta parte da dissertação é explicado o funcionamento do método proposto baseado no LVQP. A primeira parte estudada do método versa a respeito da modificação da quantização do vetor LVQP, que transforma cada vetor de diferença em um conjunto de códigos únicos, o denominado *Difference Quantization Pattern* (DQP); e a segunda parte é o processo de aplicação do DQP e do LVQP na mesma base de treino. Para além desses, também é explicado o processo de geração dos histogramas gerados por cada codificação e como eles são unificados para a classificação.

3.1 LOCAL VECTOR QUANTIZATION PATTERN - LVQP

O *Local Vector Quantization Pattern* (LVQP) (PAN; FAN; ZHANG, 2015), é um método baseado no LBP com a aplicação de um técnica de clusterização denominada quantização de vetores. Para entendermos o LVQP, primeiro temos que definir o que são vetores de diferença, que são os vetores com os valores utilizados antes da binarização do LBP pela função $s(x)$. Com o conceito de quantização de vetores definido, precisamos entender como a clusterização com a quantização de vetores funciona. As próximas seções explicam como cada um dos componentes do LVQP funciona e, por final, o seu algoritmo completo.

3.2 VETOR DE DIFERENÇA

O vetor de diferença é um componente essencial para o LVQP, já que o conjunto de vetores de diferenças é o valor de entrada para a quantização de vetores. O vetor de diferença é o conjunto de valores das diferenças de todas intensidades dos pixels vizinhos em relação ao pixel central. Na Equação (3.1), temos a definição do vetor de diferença.

$$V_d = (g_0 - g_c, g_1 - g_c, \dots, g_{P-1} - g_c). \quad (3.1)$$

Na Equação (3.1), V_d é o vetor de diferença; g_c , o valor do pixel central; g_i é o i -ésimo pixel vizinho e P o valor do total de vizinhos. O vetor de diferença é considerado como um código para o LVQP. Podemos considerar cada um dos componentes do vetor de diferença como um valor de entrada para a função de binarização na Equação (2.8) para o processamento LBP. Com a binarização algumas características podem ser perdidas tais quais a magnitude e sinal. A ausência de certas características com a binarização pode afetar o desempenho de métodos principalmente em classificações.

3.3 QUANTIZAÇÃO DE VETORES

A quantização de vetores é uma modificação do conhecido método de clusterização k -means com a adição do conceito de distorção dentro dos *clusters* gerados. Existem diversos algoritmos para quantização de vetores, cada um deles focado em uma tarefa específica. O algoritmo utilizado para quantização de vetores no LVQP é denominado Linde-Buzo-Gray (LBG) (LINDE; BUZO; GRAY, 1980). De forma geral a clusterização baseada nesse método pode ser observada no Algoritmo 1.

Algoritmo 1: Algoritmo LBG

Result: Um conjunto de centróides dos *clusters* a serem gerados (codebook)

- 1: Selecione um conjunto de dados **M** com códigos de dimensão **P**
 - 2: Defina o total de *clusters* gerados **K**
 - 3: Defina o número máximo de iterações **MaximoIteracoes**
 - 4: Defina o valor de distorção ϵ mínimo
 - 5: Defina o valor de **distorcaoAnterior**, **distorcaoAtual** e **distorcaoGeral** para 0
 - 6: Crie um novo *codebook* **C**
 - 7: Crie **N** *clusters* e para cada uma deles defina uma centróide selecionado aleatoriamente a partir do conjunto de dados
 - 8: Defina o valor de iterações para 0;
 - 9: **do**
 - for** cada código na base de dados **do**
 - Aloque o código para o *cluster* J com menor distância em relação ao centróide;
 - Atualize o valor do centróide do *cluster* J com as médias dos códigos presentes dentro do *cluster* J ;
 - end**
 - distorcaoAnterior** = **distorcaoAtual**;
 - distorcaoAtual** = 0;
 - for** cada *cluster* do conjunto de *clusters* **do**
 - distorcaoAtual** = **distorcaoAtual** + média de todas as distâncias de cada código do *cluster* em relação ao centróide;
 - end**
 - distorcaoAtual** = **distorcaoAtual** / **K**;
 - distorcaoGeral** = (**distorcaoAnterior**-**distorcaoAtual**)/**distorcaoAtual**;
 - iterações = iterações + 1;
 - 10: **while** (iterações < MaximoIteracoes) ou (distorcaoGeral > ϵ)
 - 11: Retorne a lista de centróides do *clusters* (codebook)
-

Para a atualização dos centróides a média dos componentes de cada *cluster* é calculada. A distorção de um *cluster* em um iteração i é a média da distância de todos os elementos componentes do *cluster* em relação ao centróide. A distorção geral é a medida que indica o quanto os *clusters* se modificam a cada iteração. Para o cálculo da distorção geral, consideramos todas as iterações anteriores, que pode ser observada na Equação (3.2), tal que, em qualquer iteração após a primeira, D_{i-1} é a iteração anterior, e obtemos a

distorção atual após todas as atualizações dos centróides na iteração D_i .

$$D_{geral} = (D_{i-1} - D_i)/D_i, \quad i = (1, 2, \dots, \text{MaximoIteracoes}). \quad (3.2)$$

Na Equação (3.2), i é a i -ésima iteração e MaximoIteracoes a quantidade de iterações máxima do algoritmo determinado no início do procedimento. Com uma distorção geral menor, menos transições são efetuadas nos *clusters*; quanto menor o valor da distorção geral, mais próximo do ótimo a clusterização está. Após o processo de clusterização, obtemos um conjunto de N vetores compostos pelos centróides dos *clusters*, onde cada um desses vetores tem dimensão P .

3.4 ALGORITMO - LVQP

Para o método LVQP, primeiro é preciso selecionar um conjunto de imagens para treino. A partir de cada imagem de treino, são extraídos o conjunto de todos os vetores de diferença; todos esses, de cada imagem, se tornarão a base de treino. Esse conjunto de treino gerado são os valores de entrada para a quantização de vetores.

O próximo passo para a codificação LVQP é a conversão da base de imagens. Para cada vetor de diferença da imagens de treino é efetuada uma busca de similaridade no *codebook*, considerando a distância entre o código e os centróides. A similaridade é determinada pela distância euclidiana; o novo valor do código de um determinado pixel que gerou o vetor de diferença é, então, a posição do centróide mais próximo no *codebook*. Após todas as substituições em cada imagem, o histograma de cada imagem da base é calculado para servir de entrada para algum algoritmo de classificação. A Figura 17 exemplifica esse processo: primeiro selecionamos um bloco de pixels (a), extraímos o seu vetor de diferença (b), então efetuamos a busca no *codebook* (c) e retornamos o valor da posição do centróide no *codebook* como o novo código (d).

O LVQP possui uma abordagem para o problema da rotação dos padrões. A solução consiste basicamente na rotação dos centróides presentes no *codebook* P vezes, considerando P a dimensão dos códigos gerados do vetor de diferença. Com a rotação, cada centróide tem P componentes que o descrevem; na busca pelo centróide mais próximo no *codebook*, cada componente tem P distâncias, sendo o seu menor distanciamento válido para seleção. A Equação (3.3) define como ocorrem as buscas do código com o menor valor de distância após as rotações.

$$D_{ER}(x, y) = \min[D_{euclidian}(x, y_i)], i \in (0, 1, \dots, P - 1). \quad (3.3)$$

Na Equação (3.3) nós temos que x é o centróide do *codebook*, y é o conjunto de códigos rotacionados, e P é a dimensão dos códigos. Na Figura 18, temos um exemplo de como são os códigos gerados após a rotação. No exemplo, temos um código com dimensão 8 ($P = 8$), que foi rotacionado 8 vezes, com saída de um novo conjunto de 8 códigos.

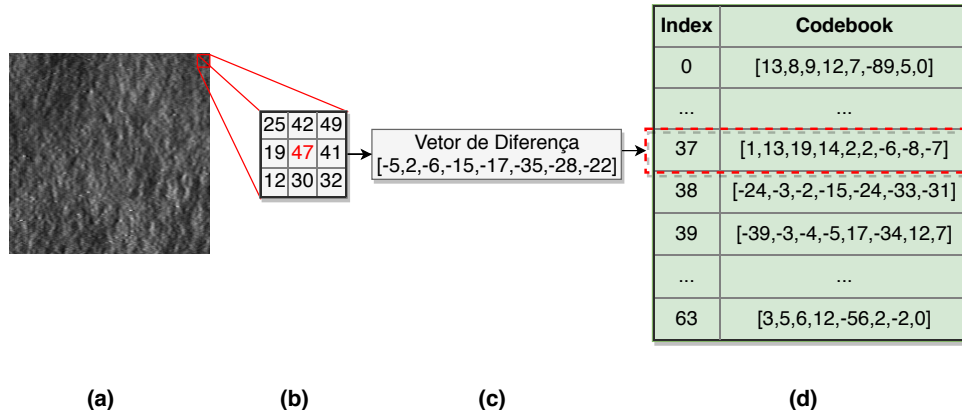


Figura 17 – Busca no *codebook* e associação no novo código LVQP. O primeiro passo é a seleção de uma textura (a), então um bloco de pixels é extraído (b); com o bloco selecionado é extraído o vetor de diferença (c); com o vetor de diferença adquirido é efetuada uma busca no *codebook* pela menor distância entre o vetor de diferença e os centróides componentes do *codebook* (d) e, finalmente, o código resultante é a posição do centróide no *codebook*, no exemplo acima é o código 37.

Id	Rotated Code
1	[X1,X2,X3,X4,X5,X6,X7,X8]
2	[X2,X3,X4,X5,X6,X7,X8,X1]
3	[X3,X4,X5,X6,X7,X8,X1,X2]
4	[X4,X5,X6,X7,X8,X1,X2,X3]
5	[X5,X6,X7,X8,X1,X2,X3,X4]
6	[X6,X7,X8,X1,X2,X3,X4,X5]
7	[X7,X8,X1,X2,X3,X4,X5,X6]
8	[X8,X1,X2,X3,X4,X5,X6,X1]

Figura 18 – Exemplo de rotação de uma código com dimensão 8. O código na posição 1 é o código original obtido do vetor de diferença, os códigos seguintes são resultados de rotação.

O vetor de diferença é capaz de armazenar mais informação do que padrões locais, considerando características com magnitude e sinal. Esse novo tratamento possibilita uma maior robustez para a possibilidade de gerar o mesmo código para padrões diferentes, que pode influenciar em um acréscimo do erro na classificação. Na Figura 19, temos um exemplo de diferentes padrões que são codificados com o mesmo código pelo método LBP, mas possuem codificação LVQP diferente.

Na Figura 19, temos, inicialmente, dois padrões diferentes; de cada padrão foi extraído o vetor de diferença, para cada vetor de diferença foi aplicada a codificação LBP e LVQP. A codificação LBP gerou o mesmo código "001100010" enquanto a codificação LVQP produz um código diferente, já que a distância entre os vetores de diferenças é considerável. Finalmente quando o histograma for computado, é mais provável o aumento de erro na classificação com a codificação LBP em relação a codificação LVQP.

Neste capítulo, podemos observar como o LVQP foi desenvolvido, e o funcionamento dele em conjunto com a aplicação à clusterização por quantização de vetores. Foram

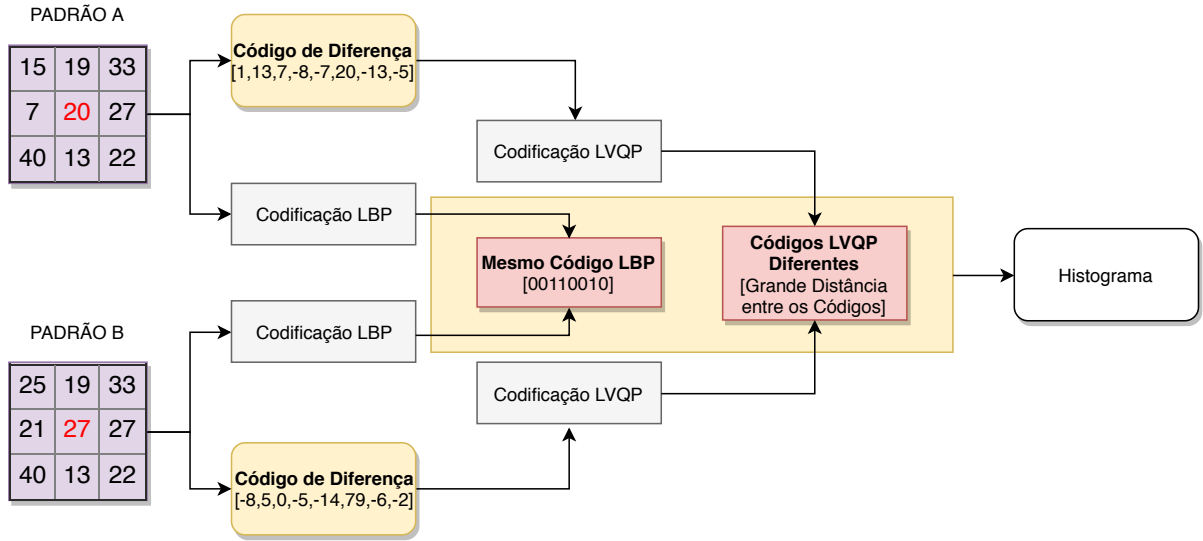


Figura 19 – LVQP x LBP. Aqui podemos observar que dois padrões diferentes produzem a mesma codificação LBP enquanto produzem codificações LVQP diferentes.

listados os novos conceitos de vizinhança e raio e como eles se aplicam ao LVQP. Obtivemos um fluxograma geral do funcionamento do LVQP e de como é efetuada sua aplicação (treino e teste).

3.5 DIFFERENCE QUANTIZATION PATTERN - DQP

Enquanto na abordagem LVQP as entradas são conjuntos de vetores de diferença, quando tratamos do *Difference Quantization Pattern* (DQP), cada vetor de diferença é um conjunto de códigos. Em DQP, um vetor de diferença com dimensão P é um conjunto de P códigos com dimensão 1. Depois de aplicar essa transformação para cada vetor de diferença extraído, obtemos um conjunto de códigos únicos. O próximo passo para o DQP é a criação de uma nova base resultante da união de todos os elementos de todos os conjunto gerados das divisões dos vetores de diferença. O novo conjunto de dados gerados pelo DQP é a entrada para a quantização de vetores. A Figura 20 demonstra como o processo de clusterização DQP ocorre. Primeiro, selecionamos um conjunto de textura para treino, de cada textura são extraídos todos os seus vetores de diferenças e convertidos para vetores com dimensão 1. No próximo passo, esses códigos são unidos em uma única base de códigos de diferença ou *Difference Codes* (DC) enviada para o algoritmo de clusterização.

Como na nova base gerada do DQP, todos os seu componentes têm dimensão 1, não é necessário a aplicação de um método invariante de rotação. Para a conversão da base, os vetores de diferenças extraídos das imagens são considerados como entradas. Com os vetores de diferenças selecionados para cada imagem, é então efetuada uma busca pelo *codebook* gerado da quantização. Para vetor de diferença extraído da base, o componente desse conjunto que apresentar menor distância em relação ao centróide é o escolhido.

Após a busca no *codebook*, o componente que possuir menor distância em relação a um

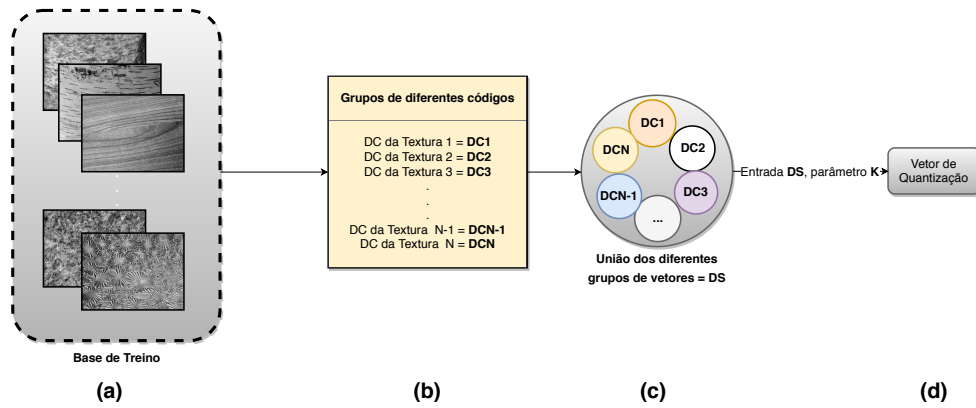


Figura 20 – Diagrama simplificado do método DQP. O processo é dividido basicamente em 4 passos: criação da base de treino com texturas de uma determinada base (a), extração de todos os vetores de diferença (b), união dos vetores de diferenças de forma que os códigos gerados tenham dimensão 1 (c) e, por fim, o envio da base gerada para a quantização de vetores (d).

determinado centróide do *codebook* tem como código a posição do centróide no *codebook* como novo valor do pixel central. Essa modificação do LVQP é o que denominamos de *Difference Quantization Pattern* (DQP). A Figura 21 exemplifica o funcionamento do DQP, desde a extração do vetor de diferença, até a busca do novo código DQP no codebook.

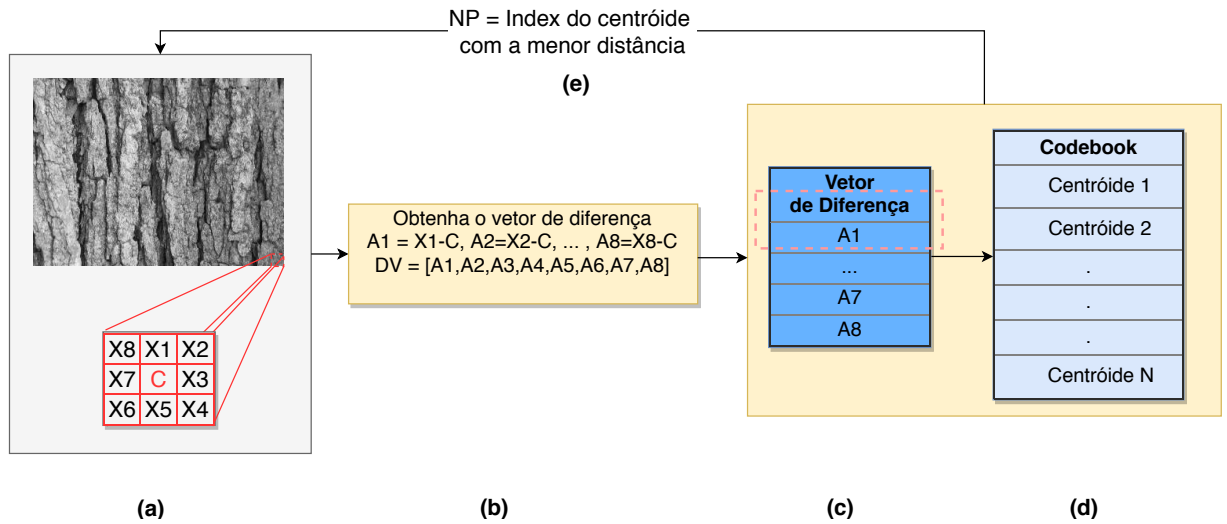


Figura 21 – Exemplo de processo de busca DQP. Primeiro iniciamos o processo com a extração do bloco de pixels de uma textura (a). A partir de um bloco de pixel, extraímos o vetor diferenças (b). Com o vetor de diferença, obtemos todas as distâncias do código gerado entre todos os componentes do *codebook* (c). Repetimos o processo para todos os códigos gerados do vetor de diferença (d) e, a partir do centróide com a menor distância encontrada dentre todos os códigos gerados do vetor de diferença, extraímos a sua posição no *codebook*, que é utilizada como um código DQP (e), substituindo o valor de intensidade do pixel em matriz nova na mesma posição que o pixel central original.

O DQP é uma modificação mais simples do LVQP. Nessa abordagem é inserida o

conceito que o vetor de diferença é um conjunto de dados individuais. Diferente do LVQP, que lida com códigos de dimensão P , o DQP lida sempre com códigos de dimensão 1; a variação do P somente afeta a dimensão final do conjunto de dados gerados da união dos vetores de diferença. O DQP é mais simples por efetuar uma busca em conjunto de dados de dimensão 1 e, por esta razão, não requer nenhuma abordagem invariante de rotação. Similar ao LVQP, os dados gerados pelo DQP são utilizados para a geração de histogramas como entrada de dados para algum método de classificação.

3.6 DIFFERENCE-LVQP - (D-LVQP)

O método LVQP apresenta excelentes resultados, com acurácias acima de 80% em bases consideradas difíceis, como a CuRET (1999). Como a proposta da manipulação dos dados gerados pelo LVQP se baseia na utilização de histogramas, é perceptível que o número de características está diretamente ligado ao número de clusters. O total de *clusters* selecionados é igual ao total de níveis do histograma extraído; se for efetuada uma clusterização com 8 componentes então é gerado um histograma com 8 níveis.

Quando o número de *clusters* selecionados para o LVQP é pequeno, durante a classificação os dados podem perder informação relevante em decorrência da grande redução de características. Na Figura 22, temos uma demonstração desse caso. Temos como entrada um conjunto de treino com 1024 características por padrão, mas, após a codificação, temos somente 8 características como entrada para o classificador.

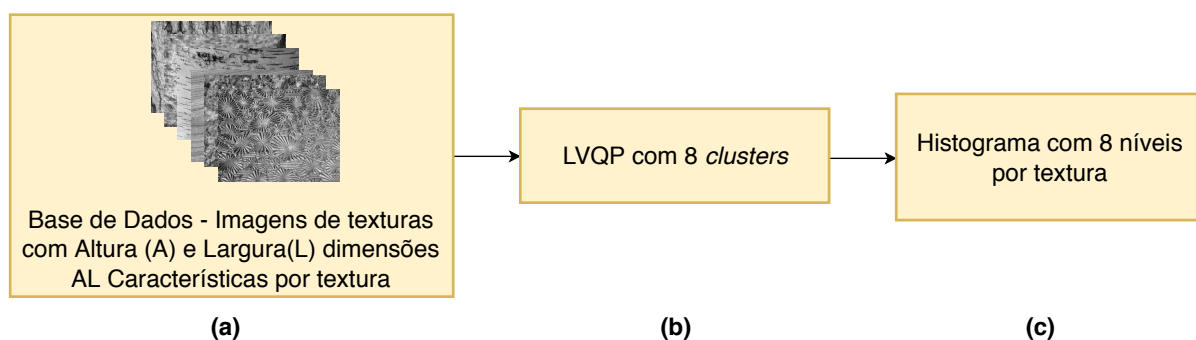


Figura 22 – Geração de histograma após abordagem LVQP. Na figura (a), temos primeiramente um conjunto de imagens com dimensão arbitrária Altura (A) e Largura (L), cada imagem tem AL características. Quando for aplicado na abordagem LVQP com 8 clusters (b), como saída teremos 8 níveis de intensidades que servem como dado para a geração dos histograma que possui igualmente 8 níveis (c).

A nossa proposta é a extração de características de cada textura, sendo essa extração efetuada pela concatenação dos histogramas gerados pelo DQP e do LVQP. Com a concatenação dos histogramas temos um acréscimo na quantidade de características. Podemos observar também que as abordagens são independentes, então podem ser efetuadas de forma simultânea. Na Figura 23, temos um exemplo de como o algoritmo funciona.

Primeiro, geramos os *codebook* LVQP e DQP das imagens de treino; após a geração convertimos as imagens em dois conjuntos de teste, utilizando um convertido com o *codebook* LVQP, e o outro com o *codebook* DQP. Finalmente, extraímos os histogramas de cada imagem convertida (DQP e LVQP) e concatenamos.

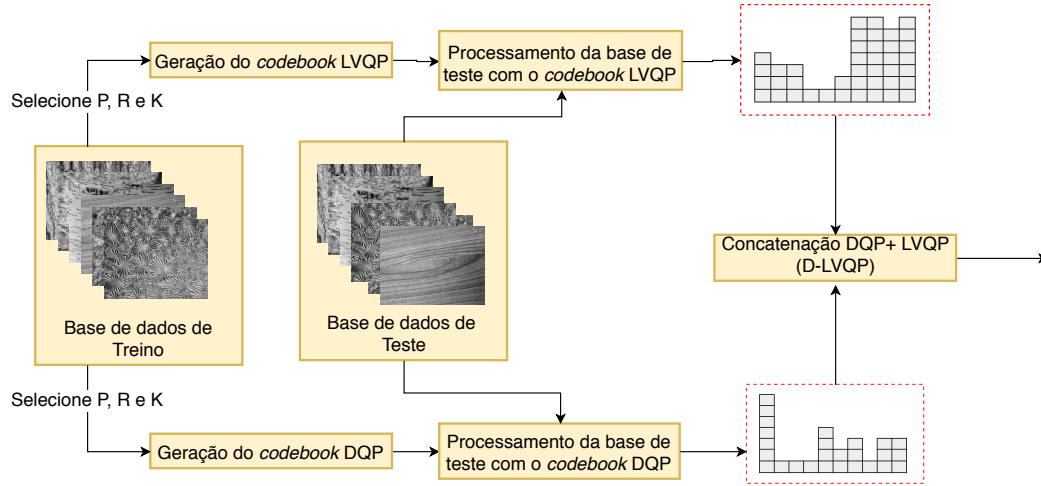


Figura 23 – Diagrama do método proposto (D-LVQP). Seguindo o exemplo, primeiro extraímos a base em treino e teste. A base de treino tem dois *codebooks*: um para o DQP e outro LVQP. A base de teste é convertida em duas codificações DQP e LVQP, das imagens de saída. De cada uma delas é computado o histograma DQP e LVQP para, finalmente, ser gerado o histograma D-LVQP(DQP+LVQP).

Na Figura 23, os métodos LVQP e DQP são executados de forma independente, compartilhando somente os atributos de entrada P, R e K . Como ambos os métodos compartilham a mesma quantidade de clusters, o número final de características sempre é $2K$. A geração dos histogramas a partir da codificação DQP em conjunto com a concatenação dos histogramas gerados pelo LVQP é o que denominamos *Difference-Local Vector Quantization Pattern* (D-LVQP). Como o DQP funciona de forma bastante similar ao LVQP, e não precisa de invariante de rotação, sua computação é efetuada de forma mais rápida. Então podemos considerar que, no pior dos casos, o tempo para o processamento do D-LVQP é o dobro de um LVQP normal.

Neste capítulo fomos introduzidos a um novo conceito o *Difference Quantization Pattern* (DQP) que, em conjunto com o *Local Vector Quantization Pattern* (LVQP) deu origem ao método proposto *Difference-Local Vector Quantization Pattern*. Podemos observar como os dois métodos DPQ e LVQP conseguem trabalhar de forma conjunta com a concatenação dos seus histogramas. Obtivemos um fluxograma completo do funcionamento do D-LVQP, desde a formação das bases de treino até a parte final de conversão e classificação.

4 RESULTADOS EXPERIMENTAIS

Nesse capítulo serão apresentados os valores padrões definidos para os experimentos e métricas para teste e treino. Posteriormente são descritos os métodos para execução do treino e teste. Aqui estão listados todos os resultados para as possíveis variações de parâmetros para as bases selecionadas. Como procedimento final efetuamos uma comparação de métodos baseados em aprendizagem profunda em relação ao método proposto.

Ainda nessa capítulo são avaliados os resultados alcançados para cada uma das bases. Para cada base foi julgado a sua validade em conjunto com teste de hipótese. Para cada conjunto de experimentos, são analisados os melhores valores alcançados pelo D-LVQP e os resultados com maior ganho em relação ao LVQP.

4.1 VARIÁVEIS, DEFINIÇÕES E MEDIDAS

Para verificar a eficácia do método proposto, foram executados uma série de experimentos em cinco bases significativas para a área: CURET (DANA *et al.*, 1999b), Brodatz (BRODATZ, 1966), UIUC (LAZEBNIK; SCHMID; PONCE, 2005), KTH -TIPS (FRITZ *et al.*, 2004) e KTH-TIPS2 (MALLIKARJUNA *et al.*, 2006). Como cada base foi criada com um diferente propósito, com diferente iluminação, pontos de vista e escala, para cada base um *codebook* foi gerado. Cada experimento foi executado 100 vezes, então a acurácia média foi calculada. Nos experimentos executados, existem algumas variáveis estáticas que se mantêm durante todo o processo: o número máximo de iterações foi de 10^4 e o valor de ϵ de distorção igual a 10^{-4} , o valor de α quando necessário foi definido para 2. Para a classificação foi utilizado o classificador *k-Nearest Neighbors*, com o valor de $K = 1$. Para efeito de comparação, os resultados dos seguintes métodos foram utilizados dos experimentos original do LVQP (PAN; FAN; ZHANG, 2015): LBP(OJALA; PIETIKAINEN; HARWOOD, 1994), LTP(TAN; TRIGGS, 2010), CLBP(GUO; ZHANG; ZHANG, 2010), DLBP(LIAO; LAW; CHUNG, 2009), BRINT(PAN; LI; WU, 2018) JEMDLBP(ZHOU; YANG; ZHANG, 2017). O método de origem LVQP(PAN; FAN; ZHANG, 2015) juntamente com o D-LVQP foram implementados com *codebook* de tamanho S variável ($S = 32, 64, 128$).

Para os experimentos, a medida de dissimilaridade entre dois histogramas utilizada foi a medida de distância Chi-Quadrada (χ^2) (DODGE, 2008). A seguir, podemos observar a função que define a medida na Equação 4.1, que tem como entrada dois vetores, que no nosso caso são histogramas, e retorna o fator de dissimilaridade entre eles. Quanto menor o valor da dissimilaridade maior é a similaridade dos histogramas.

$$d_{\chi^2}(H, K) = \sum_B \frac{(h_i - k_i)^2}{h_i + k_i} \quad (4.1)$$

Na Equação 4.1, temos que H e K são histogramas distintos, tal que h_i são k_i o i -ésimo componente de cada histograma e, por fim, B , que é a intensidade máxima dos histogramas.

4.2 BASE UIUC

A base UIUC é formada por 24 classes, cada classe possui 40 padrões, sendo suas dimensões originais de 640 pixels de largura por 480 pixels de altura. A base foi redimensionada para 20% em cada uma das suas dimensões, para um melhor desempenho computacional. Para a validação foi utilizado a validação cruzada *holdout* com valores de treino N de cada classe, tal que $N = (5, 10, 15, 20)$, ou seja de cada textura são enviadas N imagens de cada classe para o treino, e o restante para o teste. A Figura 24 mostra exemplo de cada textura presente na base UIUC.

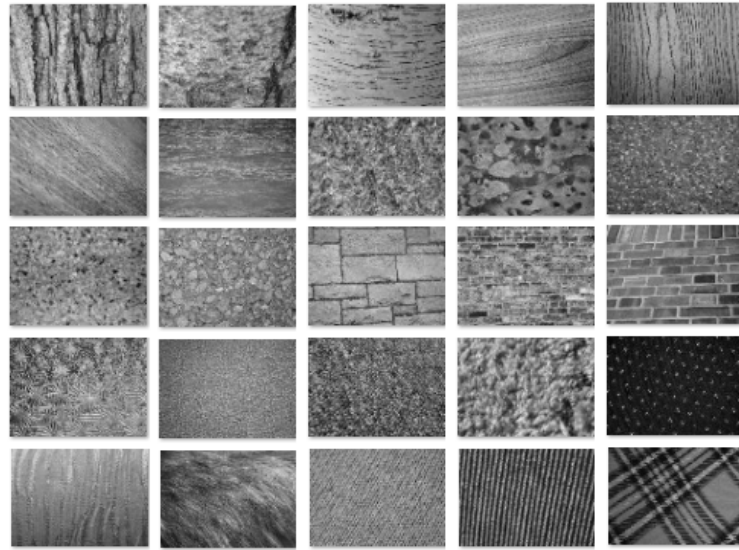


Figura 24 – Exemplo das 25 classes da base UIUC

Na Tabela 2, temos os resultados referentes à acurácia para a base UIUC com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 1$ e vizinhos $P = 8$.

Como podemos observar na Tabela 2, em todos os casos o método proposto D-LVQP obteve uma acurácia superior, sendo o seu maior valor atingido com $N = 20$ com 64 *clusters* chegando a 92,71% de acurácia. Vale ressaltar a maior diferença de ganho em relação ao método de origem com mesmos valores de *cluster* e N , ocorrendo com 128 *clusters* e $N = 5$, onde o LVQP chegou a 64,99% de acurácia enquanto o D-LVQP chegou a 75,56%, um total de 10,57 p.p. de ganho .

Na Tabela 3 temos os resultados relacionados com a acurácia para a base UIUC com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 2$ e vizinhos $P = 16$.

Tabela 2 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).

	R=1, P=8			
	20	15	10	5
LBP	54,65	52,94	47,14	39,72
LBP/ $VAR_{P,R}$	66,61	63,98	58,49	50,37
LTP	67,16	64,29	58,20	48,15
CLBP_SMC	87,64	85,70	82,65	75,05
CLBC_SMC	87,83	85,66	82,35	74,57
BRINT ^{mr1}	87,94	85,89	82,46	72,74
LVQP(32)	87,96(1,30)	82,89(1,26)	75,93(1,42)	63,64(1,96)
LVQP(64)	88,22(1,36)	83,62(1,18)	76,70(1,49)	63,98(2,03)
LVQP(128)	88,71(1,36)	84,40(1,30)	77,58(1,52)	64,99(1,91)
D-LVQP(32)	92,50(1,14)	89,18(1,10)	83,56(1,47)	73,55(2,00)
D-LVQP(64)	92,71(1,07)	89,44(1,09)	84,01(1,23)	73,59(1,59)
D-LVQP(128)	92,66(0,97)	89,60(1,01)	84,45(1,32)	75,56(1,84)

Tabela 3 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).

	R=2, P=16			
	20	15	10	5
LBP	61,32	56,42	51,16	42,03
LBP/ $VAR_{P,R}$	73,31	70,58	66,04	57,03
LTP	79,25	75,80	70,77	60,34
CLBP_SMC	91,04	89,42	86,29	78,57
CLBC_SMC	91,04	89,66	86,63	79,48
BRINT ^{mr1}	90,98	89,62	86,65	80,29
LVQP(32)	90,99(1,17)	87,15(1,20)	81,03(1,43)	70,53(1,94)
LVQP(64)	92,86(1,06)	89,23(1,19)	84,22(1,21)	74,10(1,91)
LVQP(128)	93,14(1,20)	89,91(1,02)	84,87(1,32)	75,60(1,90)
D-LVQP(32)	94,00(0,91)	90,91(0,99)	86,14(1,30)	77,27(1,73)
D-LVQP(64)	94,67(0,89)	92,29(1,02)	88,19(1,19)	78,71(1,62)
D-LVQP(128)	95,01(0,89)	92,78(0,96)	88,58(1,33)	80,55(1,62)

Como foi observado na Tabela 3, o D-LVQP obteve um ótimo despenho, já que, em todos os casos apresentou acurácia superior aos outros métodos testados. A maior acurácia do D-LVQP foi alcançada com o valor de $N = 20$ com um total de 128 *clusters* chegando a 95,01% de acurácia. Nesse experimento a maior diferença de resultados foi do D-LVQP de $N = 5$ e com 128 *cluster* com D-LVQP chegando a 80,55% em relação ao LVQP 75,60%,

com 4,95 p.p. de ganho.

Na Tabela 4, temos os resultados referentes à acurácia para a base UIUC com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 3$ e vizinhos $P = 24$.

Tabela 4 – Classificação em relação à acurácia (%) para a base UIUC com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).

	R=3, P=24			
	20	15	10	5
LBP	64,67	60,05	54,25	44,59
LBP/ $VAR_{P,R}$	75,01	71,94	66,86	57,55
LTP	82,34	79,10	73,94	62,19
CLBP_SMC	91,19	89,21	85,95	78,05
CLBC_SMC	91,39	90,10	86,45	79,75
BRINT ^{mr1}	91,94	90,62	87,68	80,22
LVQP(32)	91,19(1,10)	87,48(1,27)	81,81(1,30)	72,15(1,76)
LVQP(64)	92,47(0,98)	89,16(1,21)	84,28(1,32)	74,66(1,78)
LVQP(128)	93,38(1,13)	90,37(1,10)	86,10(1,25)	76,82(1,87)
D-LVQP(32)	93,76(0,98)	91,02(0,96)	86,67(1,31)	77,66(1,82)
D-LVQP(64)	92,57(1,09)	89,14(1,04)	84,15(1,47)	74,35(1,59)
D-LVQP(128)	95,35(0,81)	92,77(0,98)	88,83(1,27)	81,21(1,75)

No último experimento efetuado nessa base, o D-LVQP continuou apresentando o melhor desempenho. Como é observado na Tabela 4 a maior acurácia atingida com $N = 20$ com 128 *clusters* chegando a 95,35%. A maior diferença de acurácia entre os métodos LVQP e D-LVQP ocorre com 128 *clusters* e $N = 5$, onde o LVQP chegou a 76,82% de acurácia enquanto o D-LVQP chegou a 81,21%, um ganho de 4,39 p.p..

4.3 BASE CURET

A base CuRET(DANA *et al.* , 1999b) é formada por 61 classes, cada uma delas possui um total de 118 padrões, sendo utilizado para o experimento um recorte do centros das texturas com dimensões de 200 pixels de largura por 200 pixels de altura. Para a validação, foi utilizada a validação cruzada *holdout*, com $N = (6, 12, 23, 46)$, ou seja, de cada textura são enviadas N imagens para o treino e o restante para o teste. A Figura 25 tem um exemplo de algumas texturas presentes na base.

Na Tabela 5, temos os resultados relacionados com a acurácia para a base CuRet com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP, variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 1$ e vizinhos $P = 8$.

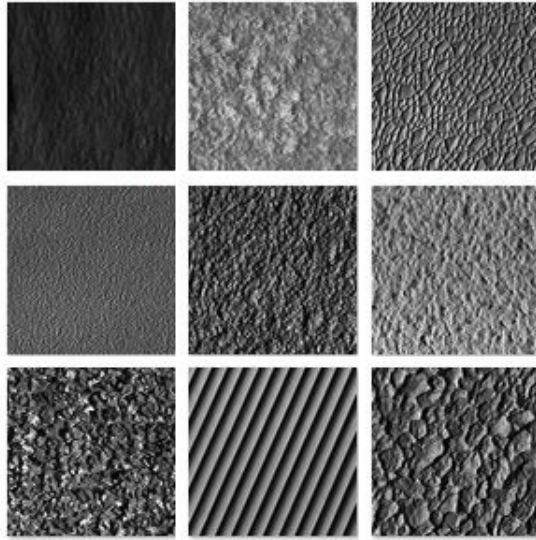


Figura 25 – Exemplos das bases da base CuRet

Tabela 5 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).

	R=1, P=8			
	46	23	12	6
LBP	90,43(0,19)	82,13(0,13)	73,83(0,90)	64,26(0,60)
LTP	86,74	80,07	71,63	62,01
VAR	68,60	60,99	52,78	43,50
CLBP_SMC	95,94	91,93	85,18	75,13
CLBC_SMC	96,02	91,06	84,44	73,65
LVQP(32)	89,63(0,52)	79,4(0,63)	67,97(0,85)	54,31(1,2)
LVQP(64)	88,59(0,57)	78,58(0,66)	67,94(0,97)	55,11(1,03)
LVQP(128)	88,38(0,49)	79,35(0,63)	69,29(0,9)	57,33(1,06)
D-LVQP(32)	91,37(0,49)	81,37(0,64)	70,02(0,82)	57,01(0,94)
D-LVQP(64)	93,02(0,47)	84,35(0,63)	73,69(0,77)	60,86(1,04)
D-LVQP(128)	94,26(0,37)	86,58(0,62)	76,36(0,8)	63,77(1,06)

Na Tabela 5 o D-LVQP não apresentou a acurácia máxima do experimento, nesse caso o CLBC_SMC apresentou o melhor desempenho com 96,02% para $N = 46$. Em todos os casos o D-LVQP apresentou melhor desempenho em relação ao LVQP. O maior ganho do D-LVQP em relação ao LVQP ocorre com 128 *clusters* e $N = 23$, onde o LVQP chegou a 79,35% de acurácia enquanto o D-LVQP chegou a 86,58%, com 7,23 p.p. de ganho.

Na Tabela 6 temos os resultados referentes à acurácia para a base CuRet com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 2$ e vizinhos $P = 16$.

Tabela 6 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).

	R=2, P=16			
	46	23	12	6
LBP	82,03(0,36)	73,48(0,20)	64,64(0,13)	54,03(0,08)
LTP	92,44	86,41	78,86	70,15
VAR	58,17	50,73	43,49	35,83
CLBP_SMC	96,13	91,52	83,97	72,64
CLBC_SMC	96,07	91,42	85,75	74,01
LVQP(32)	89,79(0,5)	79,29(0,62)	67,66(0,89)	53,68(1,21)
LVQP(64)	89,13(0,53)	79,62(0,66)	68,77(0,93)	55,87(0,97)
LVQP(128)	89,22(0,57)	80,42(0,63)	70,35(0,94)	57,85(1,09)
D-LVQP(32)	93,22(0,44)	84,35(0,62)	73,38(0,82)	59,95(1,14)
D-LVQP(64)	94,48(0,44)	86,6(0,74)	76,25(0,85)	63,02(1,08)
D-LVQP(128)	94,8(0,46)	87,63(0,62)	77,42(0,86)	64,45(1,12)

Na Tabela 6 o D-LVQP novamente não apresentou a acurácia máxima do experimento, o CLBC_SMC foi o descritor que apresentou o melhor desempenho com 96,13% para $N = 46$. Em todos os casos o D-LVQP apresentou melhor desempenho em relação ao LVQP. Quando comparamos o método de origem LVQP com a modificação D-LVQP, obtemos a maior diferença com $N = 23$ com o LVQP chegou a 80,42% de acurácia, enquanto o D-LVQP chegou a 86,63%, com 7,21 p.p. de ganho

Na Tabela 7, temos os resultados alusivos a acurácia para a base CuRet com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 3$ e vizinhos $P = 24$.

Novamente segundo a Tabela 7, o método CLBP_SMC obteve os melhores resultados, sendo a maior acurácia alcançada com 96,12% para $N = 46$. O D-LVQP em todos os casos apresentou um ganho de acurácia se considerado o método original LVQP. Neste ultimo experimento o D-LVQP obteve o maior ganho em relação ao LVQP, com 28 *clusters* e $N = 23$, onde o D-LVQP obteve 87,43% e o LVQP chegou a 80,36% de acurácia, com 7,07 p.p. de ganho.

Podemos notar nos experimentos que o nosso descritor D-LVQP não obteve em todos os casos o melhor comportamento em relação aos outros métodos. Uma das possíveis razões é a ausência dos parâmetros originais do trabalho de origem LVQP. O D-LVQP depende diretamente do LVQP, sendo seu despenho um incremento do LVQP. No trabalho de origem, o LVQP foi o melhor descritor dentre os testados para essa mesma base CuRet, isto é, se obtivéssemos os parâmetros ótimos teríamos um acréscimo na acurácia do LVQP e consequentemente do D-LVQP.

Tabela 7 – Classificação em relação à acurácia (%) para a base CuRet com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).

	R=3, P=24			
	46	23	12	6
LBP	89,18(0,21)	81,79(0,14)	72,45(0,09)	62,14(0,06)
LTP	91,42	86,39	77,11	67,72
VAR	61,87	54,40	46,61	38,39
CLBP_SMC	96,12	91,57	84,17	73,11
CLBC_SMC	96,5	92,90	87,15	76,32
LVQP(32)	89,77(0,54)	79,81(0,61)	68,34(0,96)	54,61(1,16)
LVQP(64)	89,3(0,53)	80,02(0,62)	69,33(0,89)	56,79(1,06)
LVQP(128)	89,07(0,59)	80,36(0,63)	70,78(0,79)	58,27(1,14)
D-LVQP(32)	92,88(0,51)	84,24(0,62)	73,31(0,75)	60,39(0,94)
D-LVQP(64)	93,98(0,39)	86,34(0,55)	76,13(0,9)	63,68(1,14)
D-LVQP(128)	94,49(0,44)	87,43(0,66)	77,68(0,72)	64,94(1,12)

4.4 BASE KTH-TIPS

A base KTH-TIPS(FRITZ *et al.* , 2004) é formada por 10 classes, cada classe possui um total de 81 padrões, sendo sua dimensões originais de 200 pixels de largura por 200 pixels de altura. A base foi redimensionada para 50% do seu tamanho, em cada dimensão. Para a validação, foi utilizada a validação cruzada *holdout*, com $N = (5, 10, 20, 40)$, ou seja, de cada textura são enviadas N imagens para o treino e o restante para o teste. A Figura 26 tem um exemplo de cada textura presente na base.

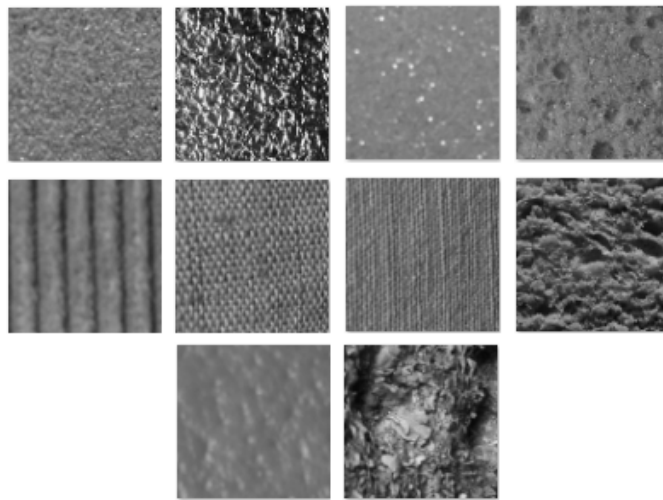


Figura 26 – Exemplo das 10 classes da base KTH-TIPS

Na Tabela 8, temos os resultados referentes à acurácia para a base KTH com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão

o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 1$ e vizinhos $P = 8$.

Tabela 8 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).

	R=1, P=8			
	40	20	10	5
LBP	93,44(1,15)	84,57(1,59)	73,49(2,32)	61,71(2,56)
LVQP(32)	91,75(1,42)	81,78(2,02)	68,21(2,04)	55,48(2,74)
LVQP(64)	93,09(1,27)	83,71(1,64)	69,71(2,37)	56,51(2,73)
LVQP(128)	93,83(1,36)	84,92(1,62)	71,49(2,56)	57,96(2,88)
D-LVQP(32)	93,72(1,29)	84,35(1,74)	71,72(1,87)	59,52(2,83)
D-LVQP(64)	94,43(1,25)	85,03(1,81)	72,48(2,08)	60,67(2,47)
D-LVQP(128)	94,86(1,24)	86,15(1,63)	72,94(2,32)	60,86(3,00)

Podemos observar que LBP descreve muito bem a base, porém o melhor resultado ainda se encontra com o D-LVQP com uma acurácia de 94,86% para $N = 40$ com 128 *clusters*. Quando comparados os métodos de origem e o nosso, o LVQP chegou a 57,96% de acurácia enquanto o D-LVQP chegou a 60,86%, sendo esse o maior ganho de 2,90 p.p. com 128 *clusters* e $N = 5$.

Na Tabela 9 temos os resultados referentes à acurácia para a base KTH com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 2$ e vizinhos $P = 16$.

Tabela 9 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).

	R=2, P=16			
	40	20	10	5
LBP	95,50(0,91)	87,00(1,92)	74,40(2,47)	61,66(2,38)
LVQP(32)	90,86(1,59)	81,08(1,79)	68,11(2,16)	55,90(2,34)
LVQP(64)	92,50(1,17)	83,63(1,79)	70,99(2,10)	58,37(2,84)
LVQP(128)	93,98(1,15)	85,76(1,64)	72,81(2,25)	59,97(2,71)
D-LVQP(32)	92,80(1,29)	82,88(1,85)	71,11(2,22)	59,16(2,91)
D-LVQP(64)	94,38(1,14)	85,23(1,89)	72,37(2,14)	60,49(2,37)
D-LVQP(128)	94,86(1,07)	86,81(1,35)	74,28(2,33)	61,88(2,81)

Na Tabela 9, o LBP conseguiu ser superior na maioria dos casos, chegando uma acurácia máxima de 95,50%. A maior diferença de ganho em relação ao método de origem LVQP chegou a 59,97% de acurácia enquanto o D-LVQP chegou a 61,88%, um ganho de 1,94 p.p., para 128 *clusters* e $N = 5$.

Na Tabela 10, temos os resultados concernentes à acurácia para a base KTH com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 3$ e vizinhos $P = 24$.

Tabela 10 – Classificação em relação à acurácia (%) para a base KTH com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).

	R=3, P=24			
	40	20	10	5
LBP	93,79(1,17)	85,67(1,61)	71,25(2,28)	57,98(2,94)
LVQP(32)	91,64(1,51)	82,33(1,79)	69,61(2,45)	57,29(2,59)
LVQP(64)	93,14(1,09)	84,47(1,79)	72,42(2,22)	59,85(2,44)
LVQP(128)	93,72(1,20)	85,56(1,75)	73,37(2,14)	61,24(2,65)
D-LVQP(32)	93,01(1,09)	83,77(1,70)	71,79(2,07)	59,82(2,73)
D-LVQP(64)	94,60(1,14)	86,12(1,53)	73,68(1,99)	61,95(3,00)
D-LVQP(128)	94,89(1,16)	87,01(1,70)	75,72(1,92)	63,12(2,95)

Como podemos observar, com o aumento do número de vizinhos e raio o D-LVQP apresentou uma melhora no comportamento, obtendo a melhor acurácia com 94,89% para $N = 40$ com 128 *clusters*. Segundo a Tabela 10 a maior diferença ocorre com 128 *clusters* e $N = 10$, nesse caso o o LVQP chegou a 73,37% de acurácia enquanto o D-LVQP chegou a 75,72%, com 2,35 p.p. de ganho.

Similar ao que ocorreu com a base CURET, o D-LVQP não apresentou o melhor desempenho em todos os casos. A base KTH-TIPS não participou dos experimentos do trabalho original, mas as mesmas suposições da falta dos parâmetros originais também se aplicam nesse caso. Quando outro método foi superior ao D-LVQP a diferença era pequena, algo que parâmetros ótimos poderiam resolver.

4.5 BASE KTH-TIPS2B

A base KTH-TIPS2(MALLIKARJUNA *et al.* , 2006) é um extensão da base CURET(DANA et al., 1999) com variação de posição, tamanho e luz, da mesma forma que a inclusão subsets de texturas existentes com outros matérias. A base possui dois conjuntos de teste, o KTH-TIPS2a e o KTH-TIPS2b, e para o experimento foi selecionado este último, que possui 11 texturas com 200 pixels de largura e 200 pixels de altura, cada textura possui 4 subconjuntos, e estes, por sua vez, têm tamanho variado de de 77 a 108 padrões.

De cada textura foi escolhido o subconjunto com o maior número de padrões (108) e efetuada um redimensionamento para 50% de cada dimensão e transformação para escala de cinza. Para o *holdout*, os valores usados de amostras foram de $N = (56, 28, 14, 7)$ imagens por classe. Na Figura 27, temos um exemplo das texturas presentes na base.



Figura 27 – Exemplos das 11 classes da base KTH-TIPS2b

Na Tabela 11, temos os resultados referentes à acurácia para a base KTH-TIPS2b com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP, variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 1$ e vizinhos $P = 8$.

Tabela 11 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 1, P = 8$).

	R=1, P=8			
	56	28	14	7
LBP	89,99(1,25)	80,87(1,30)	70,26(1,77)	59,56(2,73)
LVQP(32)	95,94(0,89)	90,08(1,15)	81,64(1,69)	72,16(2,41)
LVQP(64)	97,73(0,72)	93,06(0,94)	85,07(1,71)	75,50(2,18)
LVQP(128)	98,06(0,54)	93,55(1,07)	86,13(1,25)	76,24(2,27)
D-LVQP(32)	97,46(0,62)	93,62(0,95)	88,28(1,43)	80,63(1,96)
D-LVQP(64)	97,85(0,54)	94,16(0,88)	88,66(1,41)	81,55(2,04)
D-LVQP(128)	98,27(0,58)	94,52(0,87)	88,79(1,50)	81,42(1,97)

Na Tabela 11, o D-LVQP apresentou o melhor resultado em todos os casos, com $N = 56$ com 128 *clusters* chegando a 98,27% de acurácia. A maior diferença entre o LVQP e o D-LVQP ocorre com 64 *clusters* e $N = 7$, onde o LVQP chegou a 75,50% de acurácia enquanto o D-LVQP chegou a 81,55%, com 6,05 p.p. de ganho.

Na Tabela 12 temos os resultados alusivos à acurácia para a base KTH-TIPS2b com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 2$ e vizinhos $P = 16$.

Na Tabela 12 o D-LVQP apresentou o melhor comportamento, sendo os seus resultados superiores em todos os casos. O D-LVQP apresentou acurácia máxima com $N = 56$ com 128 *clusters* chegando a 97,82%. Se considerarmos os método de origem LVQP e o proposto D-LVQP o maior ganho de acurácia ocorre com 128 *clusters* e $N = 7$, situação

Tabela 12 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 2, P = 16$).

	R=2, P=16			
	56	28	14	7
LBP	90,24(1,08)	79,96(1,20)	68,16(1,82)	57,42(2,33)
LVQP(32)	95,49(0,76)	89,71(1,07)	81,36(1,75)	71,62(2,49)
LVQP(64)	96,49(0,78)	90,67(1,09)	82,55(1,78)	72,93(2,36)
LVQP(128)	97,18(0,67)	91,71(1,11)	83,84(1,43)	74,82(2,22)
D-LVQP(32)	97,48(0,57)	93,79(0,91)	88,57(1,55)	81,64(2,44)
D-LVQP(64)	97,68(0,57)	93,96(0,91)	88,60(1,43)	80,75(1,96)
D-LVQP(128)	97,82(0,62)	94,10(0,85)	88,51(1,50)	80,98(2,33)

esta que possibilita que o LVQP alcance 74,82% de acurácia enquanto o D-LVQP chegou a 80,96%, ou seja, 6,16 p.p. de ganho.

Na Tabela 13 temos os resultados relacionados com a acurácia para a base KTH-TIPS2b com relação aos métodos LBP e seus variantes juntamente com LVQP e D-LVQP variando de dimensão o *codebook* $S=(32, 64 \text{ e } 128)$ com raio $R = 3$ e vizinhos $P = 24$.

Tabela 13 – Classificação em relação à acurácia (%) para a base KTH-TIPS2b com diferentes métodos com raio e vizinhos fixos ($R = 3, P = 24$).

	R=3, P=24			
	56	28	14	7
LBP	91,54(1,05)	81,12(1,26)	69,97(1,84)	58,57(2,61)
LVQP(32)	95,68(0,84)	89,71(0,93)	81,50(1,61)	71,71(2,34)
LVQP(64)	96,96(0,76)	91,31(1,04)	83,49(1,50)	73,99(2,10)
LVQP(128)	97,74(0,60)	93,06(0,79)	85,50(1,36)	75,51(2,43)
D-LVQP(32)	97,29(0,57)	93,51(0,92)	87,67(1,58)	80,45(2,09)
D-LVQP(64)	97,51(0,56)	93,57(0,89)	87,89(1,51)	80,66(2,10)
D-LVQP(128)	98,02(0,62)	94,33(0,81)	88,19(1,50)	80,57(2,51)

Novamente segundo a Tabela 13, o D-LVQP demonstrou ser o melhor descritor dentre os testados, por obter em todos os casos de teste o melhor resultado. O D-LVQP alcançou a acurácia máxima com $N = 56$ com 128 *clusters* chegando a 98,02%. Levando em consideração os métodos presentes, quando comparamos os métodos de origem e o proposto, o maior ganho do D-LVQP em relação ao LVQP, ocorre com 64 *clusters* e $N = 7$, no qual o LVQP chegou a 73,99% de acurácia enquanto o D-LVQP chegou a 80,66%, que implica em 6,67 p.p. de ganho.

4.6 BASE BRODATZ

A base Brodatz(BRODATZ, 1966) é formada por 30 texturas, sendo cada uma com dimensões de 640 pixels de largura por 640 pixels de altura. De cada textura foi extraído quatro recortes de 128 de altura e 128 de largura de tal forma que não se sobrepusessem. Para a geração final da base, cada recorte foi rotacionado com passos de 15° de 0° a 180° . Obtendo uma base com 1560 imagens, com cada categoria contendo 52 imagens. Apesar da base sofrer algumas críticas a respeito da limitação de representatividade, por conter somente uma imagem de cada tipo de textura, os testes efetuados criaram novos exemplos com bons casos de variação de rotação. As imagens sem rotação foram usadas para treino que resulta em um N fixo de 4, o restante foi utilizado com teste. A Figura 28 tem alguns exemplos de texturas presentes na base.

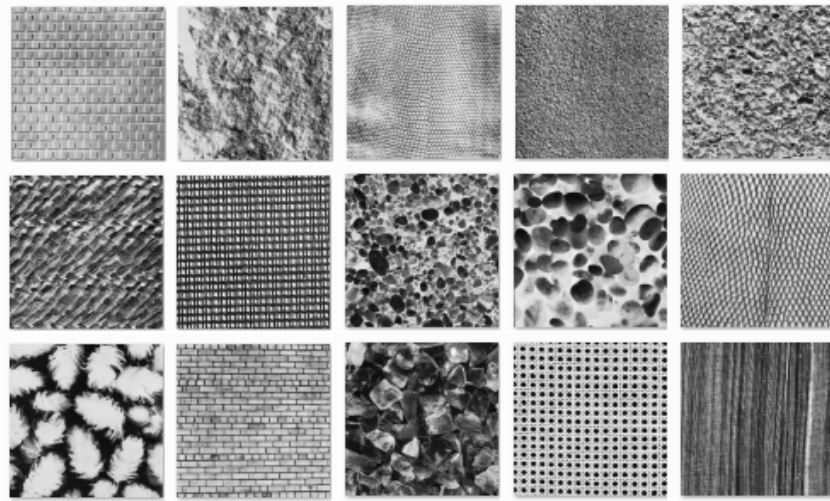


Figura 28 – Exemplos das classes da base Brodatz

Na Tabela 14 temos os resultados referentes à acurácia para a base Brodatz com relação aos métodos LBP e LVQP com variação do tamanho do *codebook* $S=(32, 64$ e $128)$.

Na Tabela 14, o EMDLBP(ZHOU; YANG; ZHANG, 2017) não possui parâmetros similares aos outros métodos (P e R), mesmo assim, ele continuou na tabela por apresentar o melhor comportamento médio. Como pode ser observado na Tabela 14, o método proposto D-LVQP obtém um desempenho superior ao método de origem LVQP, apresentando desempenho superior em todos os casos. Como o número de treino é limitado, no nosso caso $N = 4$, selecionamos o melhor caso pelos valores de P e R . A acurácia máxima do D-LVQP é de 97,92% com $R=3$, $P=24$ e $S=64$, em comparação ao LVQP que chegou a 94,58%, um ganho de 3,34 p.p..

Neste experimento o D-LVQP não obteve sempre os melhores resultados. Podemos considerar que tal desempenho ocorreu pela falta dos parâmetros do trabalho original. Com a presença dos parâmetros originais provavelmente o LVQP apresentaria um melhor

Tabela 14 – Classificação em relação à acurácia (%) para a base Brodatz com diferentes métodos

	P=1, R=8	P=2, R=16	P=3, R=24
LBP	40,28	73,19	89,65
LTP	57,50	87,78	95,76
CLBP_SMC	85,83	99,03	99,72
CLBP_SMC	89,44	97,99	98,40
JEMDLBP	-	-	92,37*
LVQP(32)	40,07	81,25	89,65
LVQP(64)	48,13	84,10	94,58
LVQP(128)	50,69	89,86	94,31
D-LVQP(32)	71,25	84,72	88,13
D-LVQP(64)	84,79	95,97	97,92
D-LVQP(128)	90,49	95,49	96,53

resultado e consequentemente o D-LVQP também apresentaria, já que o D-LVQP depende diretamente do LVQP.

4.7 MÉTODOS LBP X APRENDIZAGEM PROFUNDA

Atualmente, a área de aprendizagem de máquinas está com forte crescimento em aplicações computacionais com aprendizagem profunda. Essas aplicações com aprendizagem profunda podem resolver os mais diversos problemas, sendo alguns dos mais notáveis aqueles relacionados a visão. Considerados por muitos, a primeira aplicação em Visão computacional, LeCun (1998) sugeriu, em 1995, redes profundas convolucionais. Atualmente, a aprendizagem profunda está presente nas mais diversas áreas como reconhecimento de faces, reconhecimento de fala, segmentação, classificação, entre diversos outros(POUYANFAR *et al.* , 2018).

Das diversas abordagens de Visão Computacional com aprendizagem profunda, ainda existem poucas focadas na classificação de texturas. As poucas que existem não são denominadas redes *end to end*, ou seja, as redes trabalham basicamente como um extrator de características. Quando aplicado em redes convolucionais genéricas, os valores de classificação ainda ficam abaixo dos valores de diversos métodos LBP. Liu (LIU *et al.* , 2017) conduziu um extenso estudo sobre a taxonomia do LBP e seus derivantes, assim como comparação com métodos de aprendizagem profunda. Na Tabela 15, podemos observar a comparação dos resultados das poucas redes *end to end* com o nosso método proposto com bases que havíamos efetuado experimentos anteriormente.

Como podemos observar na Tabela 15, o D-LVQP somente perde em uma base, porém com pouca diferença. Nas outras duas bases que o método D-LVQP se apresenta mais eficiente, vale destacar o grande ganho na base KTH2-B, com um ganho de 63,66% para 98,27%. Como citado por Liu (2017), abordagens profundas possuem três grandes

Tabela 15 – Comparação em relação à acurácia de métodos de aprendizagem profunda para classificação de texturas em relação ao método proposto

	CuRet	Brodatz	KTH2-B
ScatNet	95,51	83,03	63,66
PCANet	92,03	90,89	59,43
PCANet ^{riu2}	81,48	85,76	52,15
RandNet	90,87	91,14	60,67
RandNet ^{riu2}	80,46	95,59	56,90
D-LVQP	94,49	97,92	98,27

limitantes: o tamanho da base necessária para o treino, o tempo de treino e o tempo de resposta, que por muitas vezes torna uma solução mais simples, mesmo que menos precisa, no entanto seja a mais indicada. Todo esse estudo demonstra que apesar do crescente investimento em aprendizagem profunda, a área de classificação de texturas necessita de um maior estudo para que tenhamos uma melhor compreensão a respeito do tema.

Como pode ser observado neste capítulo, foram selecionadas diversas bases de texturas relevantes para área. O D-LVQP foi comparado com diversos descritores, nas diversas bases e demonstrou ser um excelente descritor. O D-LVQP sempre se comportou de forma superior quando comparado o LVQP. O D-LVQP demonstrou ser uma saída viável em comparação com a aplicação de aprendizagem profunda para classificação de texturas.

4.8 AVALIAÇÃO DOS RESULTADOS

O D-LVQP se comportou bem em várias bases testadas, apresentando em diversos experimentos a melhor acurácia dentre os métodos selecionados. Em alguns testes, o ganho de acurácia dos D-LVQP pode ser considerado discreto, podendo, por vezes, levar ao questionamento se o método proposto realmente é melhor em relação ao método de origem. Para avaliar e validar os ganhos nos testes, mesmo em casos com ganho não tão discretos, aplicamos o teste de hipótese T-test de Welch (DERRICK; WHITE, 2016), que é indicado para testes em amostras com variâncias diferentes, que é o nosso caso. O T-teste de Welch normalmente é utilizado para a verificação se duas médias são iguais, mas no nosso caso, utilizamos para verificação se o nosso método proposto apresenta de fato uma acurácia superior.

Para a utilização do método proposto precisamos definir a hipótese nula, cuja média do nosso método D-LVQP é inferior ou igual ao método de origem LVQP. Para o nosso problema, consideramos como sendo a hipótese alternativa que o nosso método D-LVQP tem acurácia média superior ao método de origem LVQP. Podemos observar na Equação 4.2 a hipótese nula H_0 e na Equação 4.3 a hipótese alternativa H_1 .

$$H_0 : \mu_1 \leq \mu_2 \quad (4.2)$$

$$H_1 : \mu_1 > \mu_2 \quad (4.3)$$

Com as hipóteses definidas, precisamos definir a estatística t . A estatística t pode ser observada na Equação 4.4.

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\frac{S_1^2}{N_1} - \frac{S_2^2}{N_2}}} \quad (4.4)$$

Na Equação 4.4, temos que $\overline{X}_1$ e $\overline{X}_2$ são as médias da primeira e segunda amostras respectivamente, S_1^2 e S_2^2 são as variâncias da amostra, e N_1 e N_2 que são os tamanhos dessas amostragens. Como o valor de t determinado então, procuramos o seu escore para o grau de confiança α . Como valor do escore finalmente definido pela estatística t através da função de score $\phi(t)$, calculamos o p -valor, tal que p -valor = $1 - \phi(t)$. Caso o p -valor seja menor que α , a hipótese nula está invalidada, ou seja, podemos rejeitar a hipótese nula e confirmar que a média do método proposto é superior com α nível de confiança. Para todos os testes executados o nível de confiança foi $\alpha = 0,05$.

4.9 BASE UIUC

Na base UIUC, o método proposto D-LVQP se comportou bem, apresentando em todos os experimentos um desempenho superior a todos os métodos selecionados. Para confirmar-

mos a superioridade do nosso método, selecionamos os melhores resultados do D-LVQP em relação ao LVQP, com as mesmas configurações de vizinhança P , raio R e quantidade de imagens de treino N . Na Tabela 16, temos os resultados do T-test de Welch, tendo como parâmetro de avaliação o p – valor.

Tabela 16 – Comparação do método LVQP em relação o método D-LVQP para a base UIUC para o T-test de Welch

	D-LVQP	LVQP	P-VALOR
P=1,R=8	92,71(1,07)	88,22(1,36)	$\sim 9e-3$
P=2,R=16	95,01(0,89)	93,14(1,20)	$\sim 9e-3$
P=3,R=24	95,35(0,81)	93,38(1,13)	$\sim 9e-3$

Como podemos observar na Tabela 16, os valores do p – valor sempre se aproximaram de 0, ou seja, o nosso método demonstrou que em todos esse casos foi obtido um desempenho superior ao método de origem. Apesar da comprovação da eficácia do nosso método em diversificados parâmetros, as conquistas mais visíveis ocorrem quando o número de treinamento é pequeno. Na Tabela 17, temos uma relação dos ganhos de acurácia para os métodos LVQP e D-LVQP quando o número de imagem é pequeno, nesse caso o mínimo $N = 5$

Tabela 17 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 5$ para a base UIUC

	D-LVQP	LVQP	GANHO (p.p.)
P=1,R=8	75,56(1,84)	64,99(1,91)	10,75
P=2,R=16	80,55(1,62)	75,60(1,90)	4,95
P=3,R=24	81,21(1,75)	76,82(1,87)	4,39

Podemos observar na Tabela 17 um ganho contínuo na base com o valor de N baixo. O ganho de acurácia com poucas imagens é uma vantagem considerável, já que, para o treino, é necessário uma menor quantidade de imagens para uma classificação com maior ou igual acurácia, além de um menor tempo computacional.

4.10 BASE CURET

Na base CuRet, o método proposto D-LVQP se comportou bem, porém no geral inferior aos variantes do CLBP. Mesmo com o desempenho inferior aos variantes do CLBP, o D-LVQP apresentou em todos os casos um desempenho superior ao LVQP. Para confirmarmos a superioridade no nosso método, selecionamos os melhores resultados do D-LVQP em relação ao LVQP, com as mesmas configurações de vizinhança P , raio R e quantidade de imagens de treino N . Na Tabela 18 temos os resultados do T-test de Welch tendo como parâmetro de avaliação o p – valor.

Tabela 18 – Comparação do método LVQP em relação o método D-LVQP para a base CuRet para o T-test de Welch

	D-LVQP	LVQP	P-VALOR
R=1,P=8	94,26(0,37)	88,38(0,49)	$\sim 9e-3$
R=2,P=16	94,8(0,46)	89,22(0,57)	$\sim 9e-3$
R=3,P=24	94,49(0,44)	89,07(0,59)	$\sim 9e-3$

Segundo a Tabela 18, podemos observar que na base CURET o p – *valor* sempre se aproximou de 0, ou seja, o D-LVQP foi estatisticamente melhor em todos estes casos testados. Nos casos em que a base de treino era menor, os ganhos do D-LVQP em relação aos outros métodos ocorreu com o número de treinamento é pequeno.

Na Tabela 19 temos uma relação dos ganhos de acurácia para os métodos LVQP e D-LVQP quando o número de imagem é pequeno, nesse caso o mínimo $N = 6$.

Tabela 19 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 6$ para a base CuRet

	D-LVQP	LVQP	GANHO (p.p.)
P=1,R=8	63,77(1,06)	57,33(1,06)	6,44
P=2,R=16	64,45(1,12)	57,85(1,09)	6,60
P=3,R=24	64,94(1,12)	58,27(1,14)	6,67

Na Tabela 19 obtemos um acréscimo contínuo de acurácia com o valor de N mínimos. O acréscimo da acurácia com um treino com poucas imagens é uma considerável vantagem. O treino do D-LVQP necessita de uma menor quantidade de imagens para uma classificação com maior ou igual acurácia a outros métodos, além de um menor tempo computacional.

4.11 BASE KTH-TIPS

Na base KTH-TIPS, o método proposto D-LVQP apresentou uma boa acurácia, porém, no geral, inferior ao LBP. Mesmo com o desempenho inferior ao LBP, o D-LVQP apresentou em todos os casos um desempenho superior ao LVQP. Para confirmarmos a superioridade no nosso método selecionamos os melhores resultados do D-LVQP em relação ao LVQP, com as mesmas configurações de vizinhança P , raio R e quantidade de imagens de treino N . Na Tabela 20 temos os resultados do T-test de Welch tendo como parâmetro de avaliação o p – *valor*.

Na Tabela 20 o p – *valor* sempre se aproximou de 0, implicando que em todos os casos testados o D-LVQP demonstrou superioridade. Novamente a maior vantagem ocorre quando o número de treino de imagens é pequeno, nesse caso com $N = 5$.

Na Tabela 21 temos uma relação dos ganhos de acurácia para os métodos LVQP e D-LVQP quando o número de imagem é pequeno, nesse caso o mínimo $N = 5$

Tabela 20 – Comparação do método LVQP em relação o método D-LVQP para a base KTH-TIPS para o T-test de Welch

	D-LVQP	LVQP	P-VALOR
P=1,R=8	94,86(1,24)	93,83(1,36)	$\sim 9e-3$
P=2,R=16	94,86(1,07)	93,98(1,15)	$\sim 9e-3$
P=3,R=24	94,89(1,16)	93,72(1,20)	$\sim 9e-3$

Tabela 21 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 5$ para a base UIUC

	D-LVQP	LVQP	GANHO (p.p.)
P=1,R=8	60,86(3,00)	57,96(2,88)	2,9
P=2,R=16	61,88(2,81)	59,97(2,71)	1,91
P=3,R=24	63,12(2,95)	61,24(2,65)	1,88

Na Tabela 21, o D-LVQP sempre apresentou ganhos, principalmente com um número reduzido de imagens de treino N . Uma acurácia superior com poucas imagens de treino ainda se mantém com uma vantagem, além da redução do custo computacional.

4.12 BASE KTH-TIPS2B

Na base KTH-TIPS2b o método proposto D-LVQP se comportou muito bem, apresentando em todos os experimentos um despenho superior a todos os métodos selecionados. Para confirmarmos a superioridade no nosso método, selecionamos os melhores resultados do D-LVQP em relação ao LVQP, com as mesmas configurações de vizinhança P , raio R e quantidade de imagens de treino N . Na Tabela 22 temos os resultados do T-test de Welch tendo como parâmetro de avaliação o $p - valor$.

Tabela 22 – Comparação do método LVQP em relação ao método D-LVQP para a base KTH-TIPS2B para o T-test de Welch

	D-LVQP	LVQP	P-VALOR
R=1,P=8	98,27(0,58)	98,06(0,54)	$\sim 4e-3$
R=2,P=16	97,82(0,62)	97,18(0,67)	$\sim 9e-3$
R=3,P=24	98,02(0,62)	97,74(0,60)	$\sim 9e-3$

Na Tabela 22 o nosso método apresentou estatisticamente os melhores resultados, já que, o $p - valor$ sempre se aproximou de 0.

Na Tabela 23 temos uma relação dos ganhos de acurácia para os métodos LVQP e D-LVQP quando o número de imagem é pequeno, nesse caso o mínimo $N = 7$

Na Tabela 23, a base manteve um ganho percentual e relação ao LVQP em todos os casos, demonstrando que com poucas imagens o D-LVQP é superior a ainda consegue

Tabela 23 – Tabela de ganho do método D-LVQP em relação ao LVQP com poucas imagens $N = 7$ para a base KTH-TIPS2b

	D-LVQP	LVQP	GANHO (p.p.)
P=1,R=8	81,42(1,97)	76,24(2,27)	5,18
P=2,R=16	80,98(2,33)	74,82(2,22)	6,16
P=3,R=24	80,66(2,10)	73,99(2,10)	6,67

uma redução de gasto computacional.

4.13 BASE BRODATZ

A base Brodatz é um caso excepcional, pois devido ao número limitado de imagens, o experimento foi executado somente uma vez, com um valor fixo de $N = 4$. Como o experimento foi executado de forma única, não foi possível determinar uma amostra e, consequentemente, um desvio padrão para um teste de hipótese. Levando em consideração a não possibilidade do teste de hipótese, podemos avaliar os ganhos do método proposto D-LVQP em relação ao LVQP. Na Tabela 24, temos uma relação dos ganhos de acurácia para os métodos LVQP e D-LVQP.

Tabela 24 – Tabela de ganho do método D-LVQP em relação ao LVQP para a base Brodatz

	D-LVQP	LVQP	GANHO (p.p.)
P=1,R=8	90,49	50,69	39,8
P=2,R=16	95,97	84,10	11,87
P=3,R=24	97,92	94,58	3,34

Podemos observar na Tabela 19, um ganho contínuo para todos os valores, com um considerável ganho, em um caso, de mais de 30%.

Como podemos ver neste capítulo, comparamos estatisticamente todos os casos relevantes do LVQP em relação ao D-LVQP. Foram comparados os casos em que os métodos atingiram maior acurácia para cada base testada, onde o D-LVQP demonstrou superioridade. Foram comparados os métodos com a menor quantidade de imagens para treino, na qual o D-LVQP continuou demonstrando superioridade. Com os dados apresentados podemos concluir que o D-LVQP apresentou sempre um melhor comportamento em relação ao LVQP independente da configuração do ambiente de teste.

5 CONCLUSÕES E TRABALHOS FUTUROS

Como foi visto neste trabalho, o processo de classificação de textura continua sendo um problema atual e que demanda um certo trabalho para ser solucionado. Grande parte desse problema reside na dificuldade de definir o que seria textura. No decorrer da história foram criados vários conceitos para definir textura e, conseqüentemente, métodos para o tratamento dessas texturas baseadas nessas definições. Dos mais diversos métodos criados para descrever textura o que mais se destacou foi LBP. Ele obteve essa relevância por ser um método simples, rápido e eficiente, baseado fortemente na descrição estatística de textura. O LBP possui um considerável poder de descrição, porém possui diversas limitações, como sensibilidade ao ruído e à rotação. Para solucionar as limitações do LBP foram criadas diversas modificações, nas quais cada uma delas focava na resolução de algum desses problemas ou na inclusão de uma nova característica. Uma das modificações do LBP que apresentou um aumento considerável na acurácia foi a LVQP, já que ela une a abordagem LBP com a quantização de vetores.

Apesar do LVQP apresentar maior acurácia em relação ao LBP, ele ainda possui algumas limitações. Boa parte das limitações do LVQP estão relacionadas a sua redução drástica de características. Tendo este problema em vista, neste trabalho foram propostas duas abordagens: a DQP e D-LVQP. O DQP é uma modificação do LVQP, capaz de realizar quantização nas diferenças entre o pixel central e seus vizinhos. O D-LVQP é uma união do LVQP e DQP, que funciona aplicando os dois métodos simultaneamente e ao final extraíndo os histogramas de cada um, realizando sua concatenação.

O D-LVQP apresenta acurácia superior em todos os casos quando comparado ao LVQP, chegando a valores máximos para cada base: UIUC com 95,35%, KTH-TIPS2 com 98,27%, KTH-TIPS com 94,89%, CuRet 94,80% e Brodatz 97,92%. Como o resultado final deste trabalho, temos o D-LVQP sendo uma solução aceitável para a classificação de texturas, demonstrando uma acurácia superior ao seu método de origem em todos os casos de teste. Um dos pontos fortes do D-LVQP é a sua considerável capacidade de descrever texturas com um número de imagens por treino reduzido. Podemos considerar também, de forma geral, que o D-LVQP consegue lidar melhor com os limitantes de um bom descritor citado no início deste trabalho, pois cada uma das bases de teste foi gerada a partir de um conjunto de modificações de texturas, incluindo modificação de escala, rotação e ruído. A superioridade do D-LVQP em relação ao LVQP foi demonstrada em todos os casos de teste apoiados por testes de hipóteses.

Apesar do bom desempenho do D-LVQP, ele possui um tempo de computação superior em relação ao LVQP, quando aplicado de forma sequencial, ou seja, aplicando o LVQP e depois o DQP ou vice-versa. Caso uma paralelização seja efetuada com o processamento do LVQP e DQP simultaneamente, o tempo decorrido é, no máximo, o do LVQP, já durante

o processamento do DQP pula o passo de rotação de código, que o DPQ consome menos tempo de processamento por seus códigos serem unitários e o cálculo de similaridade entre eles ser extremamente simples.

Para trabalhos futuros com o D-LVQP, sugerimos paralelização do método e/ou modificação do método de clusterização para um processamento mais rápido. A concatenação com outras características pode ser uma abordagem viável. A inserção de alguma abordagem profunda em conjunto com o D-LVQP pode trazer bons frutos. Uma das limitações do nosso método foi a ausência dos parâmetros originais do método de origem LVQP, que ocasionou, em algumas bases, resultados diferentes do artigo original. A falta desses parâmetros força a utilização de parâmetros padrões que podem não ser o ótimo e, talvez, prejudicar o D-LVQP que depende do LVQP. Conseguir esses parâmetros de origem com os autores originais poder ser uma atividade futura, em busca do ótimo possível para o D-LVQP.

REFERÊNCIAS

- BABAGHORBANI, P.; PARVANEH, S.; GHASSEMI, A.; MANSHAI, K. Sonography images for breast cancer texture classification in diagnosis of malignant or benign tumors. In: *2010 4th International Conference on Bioinformatics and Biomedical Engineering*. [S.l.: s.n.], 2010. p. 1–4.
- BRODATZ, P. *Textures: a photographic album for artists and designers*. [S.l.]: Dover Pubns, 1966.
- BRYNOLFSSON, P.; NILSSON, D.; TORHEIM, T.; ASKLUND, T.; KARLSSON, C. T.; TRYGG, J.; NYHOLM, T.; GARPEBRING, A. Haralick texture features from apparent diffusion coefficient (adc) mri images depend on imaging and pre-processing parameters. *Scientific Reports*, v. 7, n. 1, p. 4041, 2017.
- COGGINS, J. M. *A Framework for Texture Analysis Based on Spatial Filtering*. Tese (Doutorado), East Lansing, MI, USA, 1983.
- DANA, K. J.; GINNEKEN, B. van; NAYAR, S. K.; KOENDERINK, J. J. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, ACM, New York, NY, USA, v. 18, n. 1, p. 1–34, jan. 1999.
- DERRICK, B.; WHITE, P. Why welch’s test is type i error robust. *The Quantitative Methods for Psychology*, TQMP, v. 12, n. 1, p. 30–38, 2016.
- DIXIT, A.; HEGDE, N. P. Image texture analysis - survey. In: *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*. [S.l.]: IEEE, 2013.
- DODGE, Y. *The Concise Encyclopedia of Statistics*. [S.l.: s.n.], 2008. ISBN 978-0-387-31742-7.
- FRITZ, M.; HAYMAN, E.; CAPUTO, B.; EKLUNDH, J. olof. *THE KTH-TIPS database*. 2004.
- GARCIA, F.; CERVANTES, J.; LOPEZ, A.; ALVARADO, M. Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets. *IEEE Latin America Transactions*, v. 14, n. 7, p. 3434–3443, July 2016.
- GONZALEZ, R.; WOODS, R. E. *Digital Image Processing*. [S.l.: s.n.], 2006. ISBN 013168728X.
- GUO, Z.; ZHANG, L.; ZHANG, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, v. 19, n. 6, p. 1657–1663, June 2010.
- HARALICK, R.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE IEEE Transactions on Systems Man and Cybernetics*, SMC-3, p. 610–621, 01 1973.
- HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, May 1979.

- HAWKINS. Textural properties for pattern recognition. *Picture Processing and Psychopictorics*, 1969.
- HE, D. chen; WANG, L. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, v. 28, n. 4, p. 509–512, July 1990.
- HUSSAIN, S. U.; TRIGGS, B. Visual Recognition using Local Quantized Patterns. In: FITZGIBBON SVETLANA LAZEBNIK, P. P. Y. S. A.; SCHMID, C. (Ed.). *ECCV 2012 - 12th European Conference on Computer Vision*. Florence, Italy: Springer, 2012. (Lecture Notes in Computer Science, v. 7573), p. 716–729.
- IEEE. Standard Glossary of Image Processing and Pattern Recognition Terminology. *IEEE Std 610.4-1990*, 1990.
- ITO, R. H.; KIM, H. Y.; SALCEDO, W. J.; PAULO, U. D. S. Classificação de Texturas Invariante a Rotação Usando Matriz de Co-ocorrência. *8th International Information and Telecommunication Technologies Symposium*, 2009.
- JULESZ, B. Visual pattern discrimination. *IRE Transactions on Information Theory*, v. 8, n. 2, p. 84–92, February 1962.
- JULESZ, B. Experiments in the visual perception of texture. *Scientific American*, Scientific American, a division of Nature America, Inc., v. 232, n. 4, p. 34–43, 1975.
- LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 8, p. 1265–1278, Aug 2005.
- LECUN, Y.; BENGIO, Y. The handbook of brain theory and neural networks. In: ARBIB, M. A. (Ed.). Cambridge, MA, USA: MIT Press, 1998. cap. Convolutional Networks for Images, Speech, and Time Series, p. 255–258. ISBN 0-262-51102-9.
- LEE, B. A new method for classification of structural textures. *International Journal of Control, Automation, and Systems*, v. 2, n. 1, p. 125–133, 2004.
- LEE, S. H.; CHOI, J. Y.; RO, Y. M.; PLATANIOTIS, K. N. Local color vector binary patterns from multichannel face images for face recognition. *IEEE Transactions on Image Processing*, v. 21, n. 4, p. 2347–2353, April 2012.
- LI, S. Z. *Markov Random Field Modeling in Image Analysis*. [S.l.]: Springer London, 2009.
- LIAO, S.; LAW, M. W. K.; CHUNG, A. C. S. Dominant local binary patterns for texture classification. *IEEE Transactions on Image Processing*, v. 18, n. 5, p. 1107–1118, May 2009.
- LINDE, Y.; BUZO, A.; GRAY, R. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, v. 28, n. 1, p. 84–95, Jan 1980.
- LIU. Extended local binary patterns for texture classification. *Image Vision Comput.*, Butterworth-Heinemann, Newton, MA, USA, v. 30, n. 2, p. 86–99, fev. 2012.
- LIU, L.; FIEGUTH, P.; GUO, Y.; WANG, X.; PIETIKÄINEN, M. Local binary features for texture classification: Taxonomy and experimental study. *Pattern Recognition*, v. 62, p. 135 – 160, 2017.

- MÄENPÄÄ, T. The local binary pattern approach to texture analysis - extensions and applications. In: . [S.l.: s.n.], 2003.
- MALLIKARJUNA, P.; TARGHI, A.; FRITZ, M.; HAYMAN, E.; CAPUTO, B.; EKLUNDH, J.-O. The kth-tips2 database. 07 2006.
- MANDELBROT, B. B. *The Fractal Geometry of Nature*. [S.l.: s.n.], 1983. v. 51. 468 p. p. ISBN 0716711869.
- MATERKA, A.; STRZELECKI, M. Texture analysis methods - a review. *COST B11 report*, 01 1998.
- NANNI, L.; LUMINI, A.; BRAHNAM, S. Survey on lbp based texture descriptors for image classification. *Expert Systems with Applications*, v. 39, n. 3, p. 3634 – 3641, 2012.
- OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In: *Proceedings of 12th International Conference on Pattern Recognition*. [S.l.: s.n.], 1994. v. 1, p. 582–585 vol.1.
- 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*, v. 24, n. 7, p. 971–987, July 2002.
- OJALA, T.; PIETIKÄINEN, M. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, v. 32, n. 3, p. 477 – 486, 1999.
- PAN, Z.; FAN, H.; ZHANG, L. Texture classification using local pattern based on vector quantization. *IEEE Transactions on Image Processing*, v. 24, n. 12, p. 5379–5388, Dec 2015.
- PAN, Z.; LI, Z.; WU, X. A new encoding scheme of lbp based on maximum run length of state “1” for texture classification. *Multimedia Tools and Applications*, Mar 2018.
- PIETIKÄINEN, M.; ZHAO, G. Two decades of local binary patterns: A survey. *CoRR*, abs/1612.06795, 2016.
- POUYANFAR, S.; SADIQ, S.; YAN, Y.; TIAN, H.; TAO, Y.; REYES, M. P.; SHYU, M.-L.; CHEN, S.-C.; IYENGAR, S. S. A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 51, n. 5, p. 92:1–92:36, set. 2018.
- QI, X.; XIAO, R.; LI, C. G.; QIAO, Y.; GUO, J.; TANG, X. Pairwise rotation invariant co-occurrence local binary pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 11, p. 2199–2213, Nov 2014.
- ROHIT, V.; ALI, J. A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 3, n. 10, p. 2277–128, 2013.
- SAROJA, G. A. S.; SULOCHANA, C. H. Texture analysis of non-uniform images using glcm. In: *2013 IEEE Conference on Information Communication Technologies*. [S.l.: s.n.], 2013. p. 1319–1322.

Comput. Vision Graph. Image Process., Academic Press Professional, Inc., San Diego, CA, USA, v. 45, n. 2, 1989.

SKLANSKY, J. Image segmentation and feature extraction. *Systems, Man and Cybernetics, IEEE Transactions on*, v. 8, p. 237 – 247, 05 1978.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. 1st. ed. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 1848829345, 9781848829343.

TAMURA, H.; MORI, S.; YAMAWAKI, T. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 8, n. 6, p. 460–473, June 1978.

TAN, X.; TRIGGS, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, v. 19, n. 6, p. 1635–1650, June 2010.

TUCERYAN, M.; JAIN, A. K. *TEXTURE ANALYSIS*. [S.l.: s.n.], 1993. 235-276 p.

WECHSLER, H. Texture analysis — a survey. *Signal Processing*, v. 2, n. 3, p. 271 – 282, 1980.

XIE, X.; MIRMEHDI, M. *A Galaxy of Texture Features*. [S.l.: s.n.], 1998. 375-406 p.

ZHOU, D.; YANG, D.; ZHANG, X. Exploring joint encoding of multi-direction local binary patterns for image classification. *Multimedia Tools and Applications*, Nov 2017.

ZUCKER, S.; KANT, K. Multiple-level representations for texture discrimination. In: *Conference on Pattern Recognition and Image Processing, Dallas, TX*. [S.l.: s.n.], 1981. p. 609–614.