



Pós-Graduação em Ciência da Computação

MARCUS VINÍCIUS SILVA LACERDA DE SOUZA

Modelo de similaridade de texturas usando Redes Neurais Profundas



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

Recife
2019

MARCUS VINÍCIUS SILVA LACERDA DE SOUZA

Modelo de similaridade de texturas usando Redes Neurais Profundas

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

Área de Concentração: Visão Computacional e Redes Neurais Profundas.

Orientador: Prof. Dr. Tsang Ing Ren

Recife
2019

Catálogo na fonte
Bibliotecária Mariana de Souza Alves CRB4-2106

S729m Souza, Marcus Vinícius Silva Lacerda de.
Modelo de similaridade de texturas usando Redes Neurais Profundas/ Marcus Vinícius Silva Lacerda de Souza. – 2019.
86 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 e Redes Neurais Profundas. 2. Análise de Texturas. 3. Classificação. 4. Deep Metric Learning. I. Ren, Tsang Ing (orientador). II. Título.

006.37

CDD (22. ed.)

UFPE-MEI 2019-156

Marcus Vinícius Silva Lacerda de Souza

“Modelo de similaridade de texturas usando Redes Neurais Profundas”

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: 30 de agosto de 2019.

BANCA EXAMINADORA

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática/UFPE

Prof. Dr. Juliano Bandeira Lima
Departamento de Eletrônica e Sistemas/UFPE

Prof. Dr. Tsang Ing Ren
Centro de Informática/UFPE
(Orientador)

Este trabalho é dedicado aos meus pais Mônica e Pedro.

AGRADECIMENTOS

Gostaria de agradecer à minha família pelo apoio em todos os momentos dessa caminhada. Ao meu orientador Tsang pelas instruções e compreensão ao longo desse trabalho. A Anna Carolina pelo apoio incondicional. A José Ivson e à toda a equipe do laboratório OKI Brasil CIn UFPE pela interminável disposição a ajudar quando preciso.

RESUMO

Nas últimas décadas, diversas técnicas de descrição de texturas foram propostas e em combinação com métodos de classificação, são utilizadas para a tarefa de análise textural. Contudo, dada a variação das classes texturas que existem, sempre que há um novo conjunto de dados são executados diversas técnicas de descrição para saber qual é a melhor nesse conjunto de dados mais recente. Encontrar as técnicas que sejam eficientes para cada uma dessas novas bases é um trabalho exaustivo, devido ao processamento computacional alto além do tempo que é utilizado no processo. O objetivo dessa dissertação é propor uma metodologia baseada em *Deep Metric Learning* que gere um espaço para representar as bases de textura de acordo com suas similaridades e com isso analisar se os métodos de descrição tem um desempenho semelhante nas bases que forem consideradas similares. Com isso, os resultados podem ser utilizados para predizer quais técnicas funcionam melhor em uma nova base de texturas de entrada que seja desconhecida ao treinamento, também otimizando o tempo e esforço que normalmente são dedicados fazendo uma análise dessa natureza. Esse espaço foi gerado aplicando uma rede profunda com uma função de perda *Triplet Loss* em bases de textura comumente utilizadas na literatura, onde já se sabe quais descritores funcionam melhor em cada uma. Como resultado, o espaço aglomerou bases semelhantes e, ao introduzirmos uma base de dados desconhecida essa base foi classificada com uma região do espaço onde ela possui maior similaridade. A análise revelou que as bases semelhantes realmente possuem uma quantidade alta de descritores de textura que são as técnicas de melhor desempenho. Os resultados mostram que a rede profunda possui um bom comportamento quando é utilizada dentro do modelo de similaridade de texturas mostrando que pode agrupar bases como regiões texturais.

Palavras-chaves: Análise de Texturas. Classificação. Descrição. Deep Metric Learning. Similaridade.

ABSTRACT

In the last decades, several texture description techniques have been proposed and in combination with classification methods are used for the textural analysis task. However, given the variation of the texture classes that exist, whenever there is a new dataset a variety of description techniques are performed to find out which one is the best in this latest dataset. Finding the techniques that are efficient for each of these new bases is exhausting work, due to the high computational processing as well as the time that is used in the process. The objective of this dissertation is to propose a methodology based on Deep Metric Learning that generates a space to represent the texture bases according to their similarities and to analyze if the description methods have a similar performance in the bases considered. similar. The results can then be used to predict which techniques work best on a new input texture base that is unfamiliar to training, while also optimizing the time and effort that is typically devoted to doing such an analysis. This space was generated by applying a deep network with a Triplet Loss function on texture bases commonly used in the literature, where it is already known which descriptors work best in each one. As a result, space has clustered similar bases, and by introducing an unknown database this base has been classified with a region of space where it has the most similarity. Analysis has shown that similar bases actually have a high amount of texture descriptors that are the best performing techniques. The results show that the deep net has a good behavior when it is used within the texture similarity model showing that it can group bases as textural regions.

Keywords: Textures Analysis. Classification. Description. Deep Metric Learning. Similarity.

LISTA DE FIGURAS

Figura 1 – Exemplos de texturas naturais.	15
Figura 2 – Exemplos de texturas artificiais.	15
Figura 3 – Exemplos de texturas artificiais usadas nos experimentos de discriminação de textura de atenção prévia.	17
Figura 4 – Exemplos de “textons” extraídos de uma imagem de textura de uma parede.	17
Figura 5 – Componentes típicos de um sistema de visão computacional.	21
Figura 6 – Um exemplo de computação GLCM em uma pequena região de textura	23
Figura 7 – Um exemplo da aplicação da transformada de Fourier em uma imagem.	24
Figura 8 – Código Gerado pelo operador LBP original.	25
Figura 9 – Código Gerado através do operador LTP.	26
Figura 10 – Efeito da rotação da imagem em uma vizinhança circular.	26
Figura 11 – Esquema do descritor BRINT.	27
Figura 12 – Imagens de entrada e suas representações através do descritor BSIF. . .	28
Figura 13 – Esquema do descritor CINIRD.	29
Figura 14 – Código CLBC.	30
Figura 15 – Framework da técnica CLBP.	31
Figura 16 – Esquema de computação FLBP.	33
Figura 17 – Esquema de computação LEP.	35
Figura 18 – Código LoQP.	36
Figura 19 – Esquema de computação MBP.	37
Figura 20 – Esquema de computação MRELBP.	38
Figura 21 – Código NRLBP.	39
Figura 22 – Código RLBP.	41
Figura 23 – Código SSLBP.	42
Figura 24 – Possíveis arquiteturas da ResNet.	43
Figura 25 – Resíduo da ResNet.	43
Figura 26 – Framework clássico da classificação de texturas.	44
Figura 27 – Problema linearmente separável e hiperplano de separação.	46
Figura 28 – Hiperplano ótimo e vetores de suporte.	46
Figura 29 – Exemplos de classes de textura da Brodatz.	47
Figura 30 – Exemplos de classes de textura da CURET.	48
Figura 31 – Exemplos de classes de textura da Outex.	49
Figura 32 – Exemplos de classes de textura das bases de dados KTH-Tips e KTH-Tips2.	50
Figura 33 – Exemplos de classes de textura da base de dados UMD.	50

Figura 34 – Exemplos de classes de textura da base de dados ALOT.	51
Figura 35 – Etapas do framework proposto.	55
Figura 36 – Dimensões do espaço de texturas reduzidas pelo t-SNE.	56
Figura 37 – Espaço textural de texturas gerado pela métrica de similaridade.	57
Figura 38 – Resnet sendo utilizada como extrator de características - a) Imagens de entrada da rede, b) Camadas da ResNet152 utilizadas para extrair o vetor de características.	60
Figura 39 – Efeito da função <i>triplet loss</i> nos elementos de treinamento.	61
Figura 40 – Arquitetura da rede <i>triplet network</i> , onde x é uma imagem da classe âncora, $x+$ um exemplo positivo dessa classe e $x-$ um exemplo negativo.	62
Figura 41 – Bases utilizadas no treinamento mostradas individualmente no espaço: ALOT, CURET, Outex e KTH.	64
Figura 42 – Espaço formado pelas bases ALOT, CURET, Outex e KTH.	65
Figura 43 – Projeção individual da base de testes Brodatz no espaço de texturas	70
Figura 44 – Distribuição da base de testes Brodatz no espaço de texturas.	71
Figura 45 – Visualização regional da distribuição da base de testes Brodatz no espaço de texturas com $k = 7$	72
Figura 46 – Histograma de classificação do k-NN da base brodatz no espaço de texturas.	73
Figura 47 – Projeção individual da base de testes UIUC no espaço de texturas.	74
Figura 48 – Distribuição da base de testes UIUC no espaço de texturas.	75
Figura 49 – Visualização regional da distribuição da base de testes UIUC no espaço de texturas com $k = 7$	76
Figura 50 – Histograma de classificação do k-NN da base UIUC no espaço de texturas.	77
Figura 51 – Projeção individual da base de testes UMD no espaço de texturas.	78
Figura 52 – Distribuição da base de testes UMD no espaço de texturas.	79
Figura 53 – Visualização regional da distribuição da base de testes UMD no espaço de texturas com $k = 7$	79
Figura 54 – Histograma de classificação do k-NN da base UMD no espaço de texturas.	80

LISTA DE TABELAS

Tabela 1 – Sumário das técnicas de descrição local utilizadas no estudo. Ruído, Rotação e Iluminação mostram se o descritor é robusto ao determinado problema. "Parcial"na coluna de ruído quer dizer que o descritor é robusto apenas aos ruídos de desfoque e ruído gaussiano.	52
Tabela 2 – Diferenças entre as bases de dados.	53
Tabela 3 – Experimento utilizando SSIM como métrica de similaridade de textura dentro de uma das 25 classes da base UMD.	59
Tabela 4 – Os melhores descritores de texturas locais baseados em sua acurácia de classificação para ALOT, Brodatz, CURET e Kth. O campo "*"significa que para a base ALOT existem apenas 15 melhores.	66
Tabela 5 – Os melhores descritores de texturas locais baseados em sua acurácia de classificação para Outex, UIUC e UMD.	67
Tabela 6 – Os piores descritores de texturas locais baseados em sua acurácia de classificação para as Bases ALOT, Brodatz, CURET e Kth. O campo "*"significa que para a base ALOT existem apenas 14 piores.	68
Tabela 7 – Os piores descritores de texturas locais baseados em sua acurácia de classificação para as Bases Outex , UIUC e UMD.	69
Tabela 8 – Melhores descritores presentes na relação ALOT e Brodatz.	73
Tabela 9 – Melhores descritores presentes na relação ALOT e UIUC.	77
Tabela 10 – Melhores descritores presentes na relação ALOT e UMD.	80

LISTA DE ABREVIATURAS E SIGLAS

2D	2 Dimensions
3D	3 Dimensions
ALOT	Amsterdam Library of Textures
CNN	Convolutional Neural Networks
CURet	Columbia-Utrecht Reflectance and Texture
DFT	Discrete Fourier Transform
GLCM	Gray Level Co-Occurrence Matrix
KNN	K Nearest Neighbor
LBP	Local Binary Patterns
LDP	Local Directional Patterns
LEP	Local Energy Patterns
LTP	Local Ternary Patterns
MRF	Markov Random Field
PCA	Principal Components Analysis
SSIM	Structural Similarity
SVM	Support Vector Machine

SUMÁRIO

1	INTRODUÇÃO	15
1.1	O QUE SÃO TEXTURAS?	15
1.2	PERCEPÇÃO HUMANA	16
1.3	ANÁLISE DE TEXTURAS E APLICAÇÕES	18
1.4	DESAFIOS	19
1.5	OBJETIVOS	19
1.6	ORGANIZAÇÃO DO TRABALHO	20
2	ESTADO DA ARTE	21
2.1	INTRODUÇÃO	21
2.2	ANÁLISE DE TEXTURAS	21
2.3	MODELO CLÁSSICO DE EXTRAÇÃO DE CARACTERÍSTICAS	21
2.3.1	Abordagens estruturais	22
2.3.2	Abordagens estatísticas	22
2.3.3	Abordagens baseadas em modelo	23
2.3.4	Abordagens baseadas em processamento de sinal	23
2.3.5	Descritores locais de textura	23
2.3.5.1	Local Binary Patterns	24
2.3.5.2	Local Ternary Patterns	25
2.3.5.3	Local Binary Patterns Fourier Histogram	26
2.3.5.4	Binary Rotation Invariant and Noise Tolerant	26
2.3.5.5	Binarized Statistical Image Features	27
2.3.5.6	Central Intensity Neighbor Intensity Radial Difference	28
2.3.5.7	Completed Local Binary Count	29
2.3.5.8	Completed Local Binary Pattern	30
2.3.5.9	Completed Local Binary Pattern Fourier Histogram	31
2.3.5.10	Discriminative Completed Local Binary Pattern	31
2.3.5.11	Dominant Local Binary Patterns	32
2.3.5.12	Extended Local Binary Patterns	32
2.3.5.13	Fuzzy Local Binary Patterns	32
2.3.5.14	Improved Local Binary Patterns	33
2.3.5.15	Local Binary Pattern Difference	33
2.3.5.16	Local Binary Pattern Variance	34
2.3.5.17	Local Energy Pattern	34
2.3.5.18	Local Frequency Descriptor	35
2.3.5.19	Local Quinary Patterns	35

2.3.5.20	Median Binary Patterns	36
2.3.5.21	Multichannel Decoded Local Binary Patterns	37
2.3.5.22	Median Robust Extended Local Binary Pattern	37
2.3.5.23	Multi-scale Joint Encoding of Local Binary Patterns	38
2.3.5.24	Noise-Resistant Local Binary Pattern	38
2.3.5.25	Noise Tolerant Local Binary Pattern	39
2.3.5.26	Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern	40
2.3.5.27	Rotation Invariant Local Phase Quantization	40
2.3.5.28	Robust Local Binary Pattern	40
2.3.5.29	Scale Selective Local Binary Patterns	41
2.3.6	Representação de texturas com Redes Convolucionais	42
2.3.6.1	ResNet	42
2.4	CLASSIFICAÇÃO DE TEXTURAS	44
2.4.1	K-Nearest Neighbor	45
2.4.2	Support Vector Machine	45
2.5	BASES DE TEXTURA	47
2.5.1	Brodatz	47
2.5.2	CUReT	48
2.5.3	Outex	48
2.5.4	KTH	49
2.5.5	UMD	50
2.5.6	ALOT	51
2.6	DISCUSSÃO	51
3	MÉTODOS PROPOSTOS	54
3.1	MOTIVAÇÃO	54
3.2	ARQUITETURA PROPOSTA	54
3.2.1	Similaridade de Texturas	58
3.2.1.1	SSIM	58
3.2.1.2	Triplet loss function e Redes Convolucionais	59
4	EXPERIMENTOS E RESULTADOS	63
4.1	INTRODUÇÃO	63
4.2	CONFIGURAÇÃO DAS BASES DE DADOS E DO EXPERIMENTO	63
4.2.1	Espaço de texturas	63
4.2.2	Classificação de teste em uma base desconhecida	64
4.3	AVALIAÇÕES	66
4.3.1	Brodatz	70
4.3.2	UIUC	73
4.3.3	UMD	78

5	CONCLUSÃO	81
5.1	TRABALHOS FUTUROS	81
	REFERÊNCIAS	82

1 INTRODUÇÃO

Dentro da visão computacional existem vários problemas que utilizam a extração de informações de imagens, sejam elas em forma estática ou em vídeos. Os avanços tecnológicos permitem que os estudos possuam uma fonte bastante diversa de dados. A extração de informação de textura das imagens é utilizada para resolver esses problemas.

1.1 O QUE SÃO TEXTURAS?

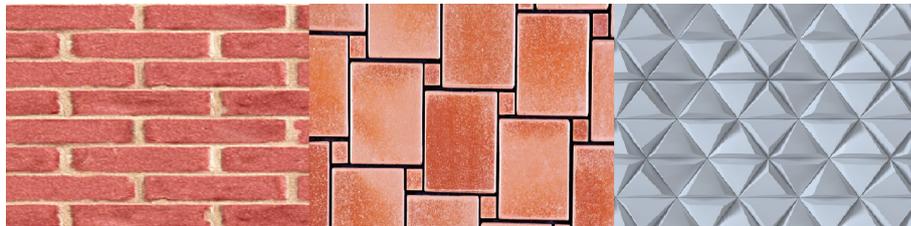
Texturas são características visuais e táteis de uma superfície (MATERKA; STRZELECKI et al., 1998); a maneira de união das partículas de um objeto ou substância. A superfície de todos os objetos que são visíveis possui uma determinada escala de textura então, por consequência, o mundo é composto uma vasta diversidade de exemplos de texturas. Além disso, texturas também podem ser geradas artificialmente aumentando a possibilidade de exemplos existentes. Existem diversas descrições para as texturas, por exemplo, texturas lisas, enrugadas, macias, ásperas e muitas outras. A Figura 1 mostra exemplos de texturas naturais e a Figura 2 mostra exemplos de texturas artificiais.

Figura 1 – Exemplos de texturas naturais.



Fonte: Google.com, 2019.

Figura 2 – Exemplos de texturas artificiais.



Fonte: Google.com, 2019.

Apesar de ser amplamente explorada dentro da área de visão computacional, a textura não possui uma única definição. Acredita-se que a textura de uma imagem é uma fonte rica em informação visual (MATERKA; STRZELECKI et al., 1998). Muitos estudiosos

deram algumas definições sobre textura, como Haralick et al. (1979) que definem textura como sendo um fenômeno de área organizado, que quando é decomposto, possui duas dimensões básicas. A primeira dimensão é para descrever as primitivas das quais a textura da imagem é composta, e a segunda é a descrição da dependência ou interação espacial entre as primitivas de uma textura de uma imagem. Outra definição de Textura dada por Castellano et al. (2004), diz que a textura de uma imagem refere-se à aparência, estrutura e arranjo das partes de um objeto dentro da imagem. Materka, Strzelecki et al. (1998) definem textura como padrões visuais complexos compostos de entidades, ou subpadrões, que possuem brilho, cor, inclinação, tamanho, etc.

De maneira geral, texturas podem ser divididas em dois tipos: texturas táteis e texturas visuais. Textura tátil se trata do ato de tocar uma superfície e a resposta imediata daquela superfície ao toque. Texturas visuais se tratam da experiência visual que uma textura produz para um observador. Comumente são relacionadas a variação espacial de estimulantes da superfície como cor, intensidade da imagem, contraste, etc.

Texturas visuais representam um importante campo dentro da visão computacional, e é ampla e historicamente explorado. Então, dentro dessa área, o termo "textura" é apenas atribuído as texturas visuais já que são o objeto principal da pesquisa. O estudo de texturas se tornou uma ótima opção para resolução de problemas dentro da visão computacional.

1.2 PERCEPÇÃO HUMANA

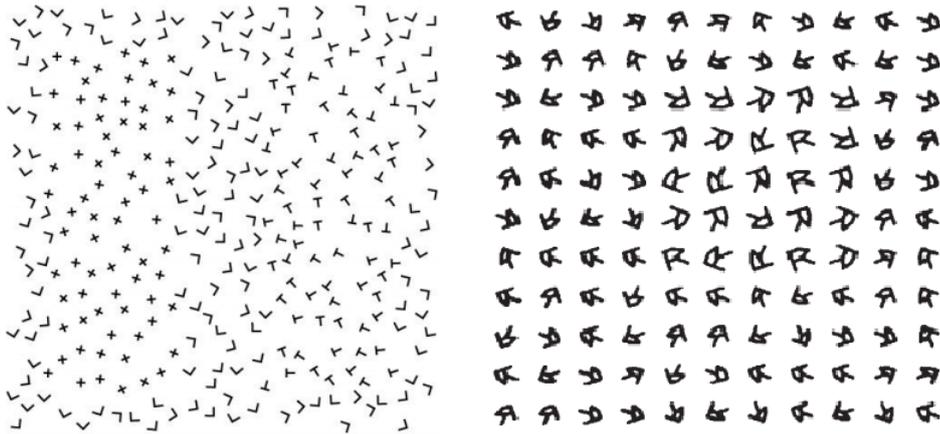
O sistema visual humano evoluiu durante milhares de anos para perceber as informações úteis que texturas trazem sobre um determinado cenário, apesar de a maioria das texturas ser de fácil percepção. Contudo, essa atividade feita automaticamente por um computador ainda é um desafio. O nível de abstração começa com a informação do *pixel*, que é a unidade mais fragmentada de uma imagem digital, segue com a primitiva da imagem (bordas, por exemplo) e a textura trazendo mais informações. Então temos região, objeto e cena. O nível de abstração exigido para análise de textura é baixo quando comparado com alguns problemas de reconhecimento de objetos e compreensão de cena, porém, não deixam de envolver dificuldades e desafios.

Pesquisas mostram que a percepção de textura na visão humana é uma das primeiras etapas de como os humanos percebem diferentes objetos em uma cena para entendê-la (JULESZ, 1962). Processos relativamente adiantados no sistema visual podem usar informações de textura para realizar uma segmentação provisória da imagem em regiões para facilitar a carga de processamento em estágios computacionais subsequentes (LANDY; GRAHAM, 2004).

A primeira estudiosa a fazer um estudo apontado para a análise de texturas foi Julesz (1962). Ela tentou determinar como o sistema visual humano de baixo nível responde à variação de uma determinada estatística de ordem. Exemplos de estatísticas de ordem

incluem contraste, homogeneidade e curvatura sendo eles de primeira, segunda, e terceira ordem respectivamente. A Figura 3 mostra imagens sintéticas usadas nos experimentos de discriminação de textura de atenção prévia (JULESZ, 1962).

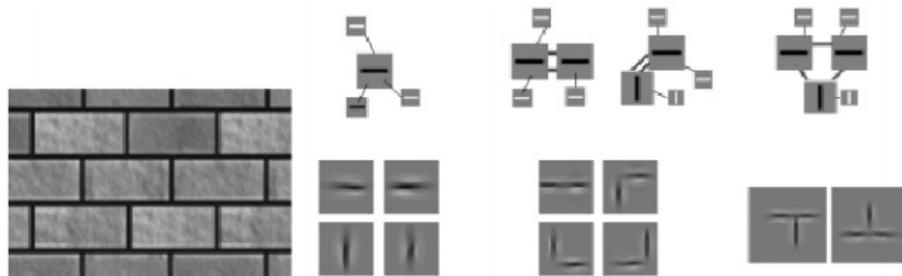
Figura 3 – Exemplos de texturas artificiais usadas nos experimentos de discriminação de textura de atenção prévia.



Fonte:(JULESZ, 1981)

Julesz (1981) também contribuiu com a “teoria dos textons”, que diz que os “textons” são as “unidades putativas de percepção de textura humana pré-atenciosa”, relacionadas às características locais da textura, como bordas, extremidades de linha, bolhas, etc. A Figura 4 mostra os “textons” de uma imagem.

Figura 4 – Exemplos de “textons” extraídos de uma imagem de textura de uma parede.



Fonte:(JULESZ, 1981)

1.3 ANÁLISE DE TEXTURAS E APLICAÇÕES

A análise de texturas é amplamente explorada devido a sua importância dentro de aplicações da computação, principalmente quando essas aplicações estão segmentando ou classificando imagens, sendo tópico de estudos desde 1962 (JULESZ, 1962). Muito disso se deve ao fato de que exemplos de imagens não param de aparecer devido aos avanços e capacidades tecnológicas. Dentro da análise de texturas existem 4 problemas principais: classificação, discriminação, extração e forma (MATERKA; STRZELECKI et al., 1998). Cada um deles trabalha de uma forma diferente de analisar as texturas de maneira correlata.

Para a resolução desses problemas, muitas vezes, envolvem o desenvolvimento de um algoritmo ou um modelo que faça a previsão automática de uma classe para uma imagem de entrada. Para isso, informações primárias importantes sobre a textura precisam ser extraídas dos *pixels* para o problema ser resolvido. Classificação de textura envolve decidir a categoria de textura à qual uma imagem observada pertence (TUCERYAN; JAIN, 1993). Isso significa que o classificador analisa as imagens que possuem rótulos como “madeira”, “pedra”, “regular”, “irregular”, entre outros. Normalmente, a classificação de texturas é abordada com vários exemplos para cada classe de textura sobre um classificador supervisionado. Exemplos da aplicação de classificação de texturas incluem Reconhecimento facial (CHAKRABORTI; CHATTERJEE, 2014), Rastreamento de objetos (TAKALA; PIETIKAINEN, 2007), Indústria (JING et al., 2012), Medicina (NAILON, 2010) e Robótica (HUGHES; CORRELL, 2015).

Na discriminação de texturas, isto é, segmentação de texturas, envolve o agrupamento de dados de imagem em regiões com características semelhantes (DERIN et al., 1984). A segmentação de texturas é um problema difícil porque geralmente não se sabe *a priori* que categorias de texturas existem em uma imagem, quantas texturas diferentes existem e quais as regiões da imagem que possuem texturas. (TUCERYAN; JAIN, 1993). Exemplos de aplicações usando segmentações de texturas incluem Medicina (MIR; HANMANDLU; TANDON, 1995), Robótica (HUGHES; CORRELL, 2015), entre outros.

A síntese de textura é uma maneira alternativa de criar texturas importante para muitas aplicações em computação gráfica, visão e processamento de imagens (MATERKA; STRZELECKI et al., 1998). Dada uma amostra de textura, o processo sintetiza uma nova textura que parece ser gerada pelos mesmos processos estocásticos subjacentes de uma textura não sintetizada. Aplicações incluem Medicina (ZHU et al., 2006) e Pintura Digital (LEWIS, 1984).

Por fim, a forma da textura é usada para reconstruir uma superfície 3D a partir de uma imagem 2D estimando a forma ou orientação com base nas propriedades da textura (MATERKA; STRZELECKI et al., 1998). A forma e a profundidade de um objeto podem ser estimadas com base na aparência visual (deformação) da textura da superfície resultante da projeção do objeto 3D no plano 2D. Os trabalhos de (CRIMINISI; ZISSERMAN, 2000) e (LOBAY; FORSYTH, 2006) mostram aplicações práticas da forma da textura.

Para todas essas tarefas a extração de características faz um papel decisivo: é nessa parte em que são extraídas as informações que vão dar suporte a resolução do problema. Historicamente (WESZKA; DYER; ROSENFELD, 1976), os métodos de extração de características utilizam a engenharia de características (*feature engineering*). Com os avanços nesse campo, metodologias de descrição local se tornaram cada vez mais eficientes (AHONEN et al., 2009).

Com o avanço do estudo das redes neurais e também da tecnologia, as redes convolucionais (CNN) começaram a ser aplicadas como métodos de extração de características, obtendo resultados tão bons ou melhores do que as metodologias clássicas (HE et al., 2016). Contudo, isso não invalida o estudo de métodos baseados em *feature engineering*: os métodos que utilizam redes profundas possuem um custo computacional alto, um tempo de extração geralmente elevado além de precisarem de uma quantidade de dados muito alta para funcionar bem (LIU et al., 2017). Em cenários em que não se tem muita informação ou poder computacional, os métodos de engenharia de características levam vantagem por serem mais simples computacionalmente, porém bem robustos em diversas variáveis.

1.4 DESAFIOS

A maioria dos métodos de análise de textura trazem a extração de características, que descrevem as propriedades de uma imagem para classificá-la. Esses métodos são chamados de descritores de características. Entretanto, como existem tantos descritores é difícil saber quando e porquê usar um determinado descritor. Conjuntos de dados são lançados frequentemente na literatura trazendo problemas dos mais variados. Quando se quer descobrir qual descritor de característica é melhor para um determinado conjunto de dados não há outra solução que não seja testar um por um naquela base. O tempo, esforço e recursos utilizados para isso podem ser consideravelmente grandes.

1.5 OBJETIVOS

O objetivo deste trabalho é propor uma metodologia baseada em *Deep Metric Learning* que gere um espaço para representar as bases de texturas de entrada através de suas similaridades. Com esse espaço de características, podemos analisar se os métodos de descrição têm um desempenho semelhante entre as bases que foram agrupadas como similares. Esse modelo será usado para prever qual técnica de representação de características texturais são boas para uma base de dados não conhecida, reduzindo o tempo e o custo de processamento da tarefa de análise de texturas. Apesar destes pontos levantados serem conhecidos na literatura, dentro do nosso conhecimento, não existe nenhum trabalho que proponha alguma metodologia para resolvê-lo.

1.6 ORGANIZAÇÃO DO TRABALHO

As seções restantes desta dissertação encontram-se estruturadas da seguinte forma:

2 Estado da Arte: apresenta os trabalhos realizados pela comunidade científica no desenvolvimento de métodos relacionados a análise de texturas.

3 Métodos propostos: descreve todos os métodos utilizados no trabalho.

4 Experimentos e Resultados: descreve e discute os resultados dos experimentos efetuados para análise e avaliação do modelo proposto.

5 Conclusão: apresenta as considerações finais sobre os principais tópicos abordados nesta dissertação, incluindo as contribuições alcançadas e as indicações de trabalhos futuros.

2 ESTADO DA ARTE

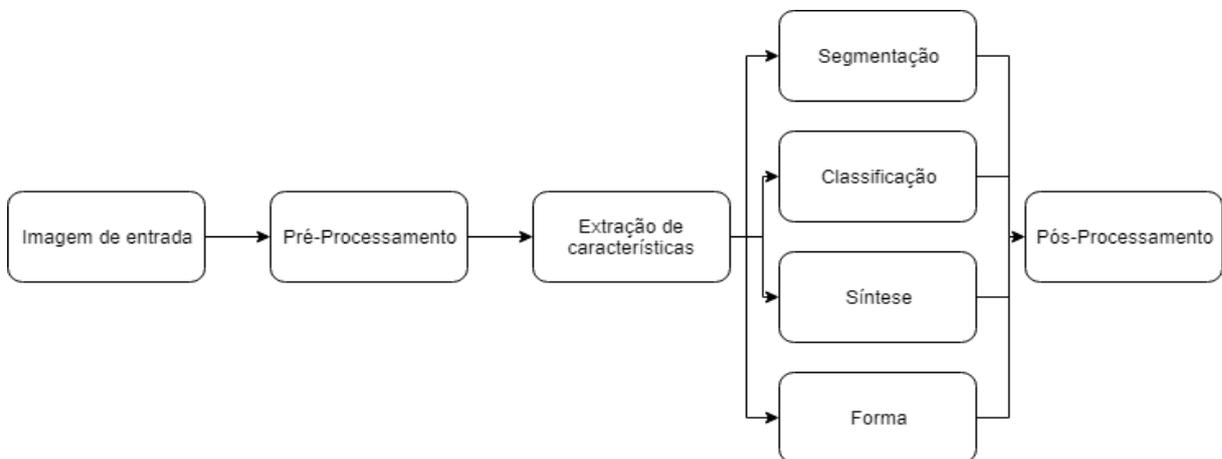
2.1 INTRODUÇÃO

Este capítulo analisa vários avanços importantes nas áreas de análise de textura, mostrando os modelos clássicos de extração de características. Também inclui métodos de descrição que são abordados no trabalho bem como as metodologias atuais de aprendizagem profunda usados em textura. Além disso, um *overview* sobre as bases de textura é apresentado.

2.2 ANÁLISE DE TEXTURAS

A extração de características tem um papel central em diversos métodos de análise de textura na visão computacional, já que tratam de obter informações sobre a textura da imagem processada. Essas informações são seguidas de métodos de classificação como o *K-Nearest Neighbors* (KNN) e o *Support Vector Machine* (SVM), como mostra a Figura 5, que utilizam essas informações de textura com o objetivo de resolver um problema.

Figura 5 – Componentes típicos de um sistema de visão computacional.



Fonte: Elaborado pelo autor, 2019.

2.3 MODELO CLÁSSICO DE EXTRAÇÃO DE CARACTERÍSTICAS

Mesmo que um classificador seja muito bom, se o método de extração de características for fraco ele vai falhar em obter um resultado aceitável. Isso acaba evidenciando o importante papel que a representação de características tem dentro da análise de texturas. As características extraídas podem posteriormente ser agrupadas, classificadas, etc., para resolver problemas de análise de textura, conforme explicado na Seção 2.2.

As 6 décadas passadas mostraram diversas categorias de técnicas e teorias dentro da extração de características de texturas. Os estudos de Haralick et al. (1979) e Weszka, Dyer e Rosenfeld (1976) mostram diversas características de textura que nasceram antes dos anos 80. Em 1996, Ojala, Pietikäinen e Harwood (1996) realizaram um estudo comparativo que mostrava os resultados da comparação de vários descritores de texturas em bases conhecidas. Em 2002, o trabalho de Zhang e Tan (2002) revisou métodos que fazem extração de características de textura que são invariantes a rotação e iluminação. O livro de Mirmehdi, Xie e Suri (2009), lançado em 2008 de título *Handbook of Texture Analysis* mostra um estudo extenso sobre análise de texturas, 2D e 3D em todos os problemas dentro deste campo de atuação.

Os métodos podem ser categorizados em estruturais, estatísticos, baseados em modelos e de processamento de sinal (MATERKA; STRZELECKI et al., 1998).

2.3.1 Abordagens estruturais

As abordagens estruturais analisam a regularidade com que as primitivas de uma textura são organizados no espaço. Esses métodos normalmente identificam tais primitivas, e em seguida inferem as regras de posicionamento das formas primitivas. A estrutura e a distribuição espacial das primitivas de textura são os principais componentes dos modelos estruturais. Essas primitivas de textura geralmente se referem a manchas, ou seja, regiões na imagem com níveis de cinza uniformes (MATERKA; STRZELECKI et al., 1998) uma vez que isso é considerado como minúcia significativa de uma imagem textural. Dada a natureza sobre as regras de posicionamento e primitiva, abordagens estruturais são melhor aplicadas quando utilizadas para texturas regulares, já que esses métodos não são projetados para lidar com texturas com um grau alto de variabilidade de padrões. Além disso, os elementos de textura também podem ser alcançados considerando medidas estatísticas (OJALA; PIETIKÄINEN; HARWOOD, 1996).

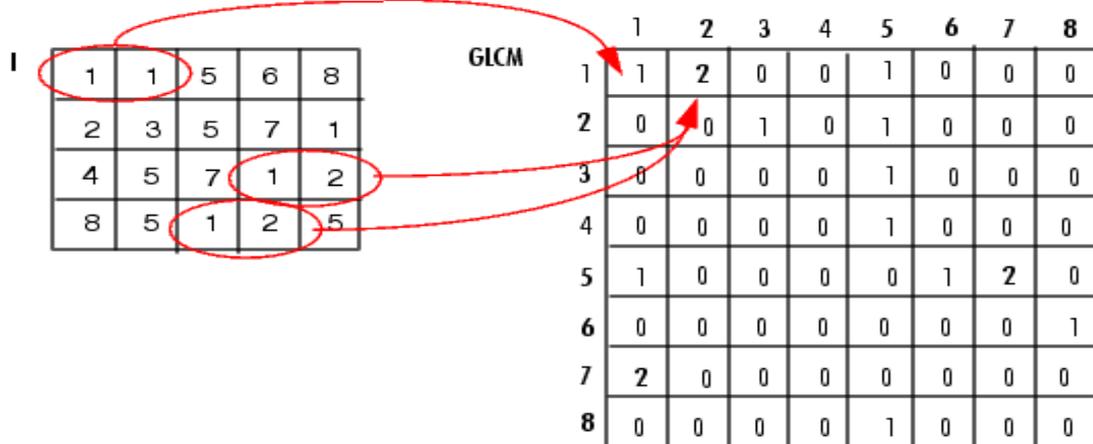
2.3.2 Abordagens estatísticas

A distribuição espacial dos valores de cinza é uma das minúcias que definem a textura (MATERKA; STRZELECKI et al., 1998). Os métodos estatísticos coletam informações de sinal de imagem do domínio espacial em forma de descritores. As abordagens estatísticas descrevem as relações entre os valores dos *pixels* com base na primeira, segunda ordem ou de ordem mais alta. Estatísticas de primeira ordem, como a média ou o desvio padrão, se referem às propriedades do *pixel* individualmente.

Haralick et al. (1979) sugeriu o uso de matrizes de co-ocorrência de nível de cinza (GLCM), que se tornou uma característica bastante conhecida da literatura.

A função de autocorrelação é outra abordagem estatística que calcula o produto escalar de uma imagem com instâncias deslocadas da mesma imagem. É normalmente usada

Figura 6 – Um exemplo de computação GLCM em uma pequena região de textura



Fonte: (MATERKA; STRZELECKI et al., 1998).

na detecção de padrões de textura repetitivos (primitivos) e descrever a regularidade e grossura das texturas (MATERKA; STRZELECKI et al., 1998).

2.3.3 Abordagens baseadas em modelo

Nas abordagens baseadas em modelo são feitas as capturas das minúcias de uma textura, configurado com parâmetros estimados. Eles são baseados na construção de um modelo de imagem que pode ser usado não apenas para descrever a textura, mas também para sintetizá-la (MATERKA; STRZELECKI et al., 1998). Entre as abordagens mais populares baseadas em modelo está o trabalho de Markov (CROSS; JAIN, 1983). Um *Markov Random Field* (MRF) pressupõe que uma intensidade de *pixel* depende apenas de seus *pixels* vizinhos, capturando restrições contextuais locais para modelar globalmente uma imagem. As principais dificuldades desses métodos são encontrar funções de energia apropriadas e conseguir otimizá-las.

2.3.4 Abordagens baseadas em processamento de sinal

Dada a evidência de que o cérebro humano faz uma análise de frequência da imagem (JULESZ, 1962), esses métodos se referem ao domínio da frequência em que as características estão relacionadas à estatística das respostas de um filtro. Abordagens de processamento de sinal se tratam de bancos de filtros, *wavelets* e transformadas de Fourier. A Figura 7 mostra um exemplo de imagem e sua representação no domínio de frequência através da transformada de Fourier.

2.3.5 Descritores locais de textura

Os estudos mais recentes de Liu et al. (2017) Huang et al. (2011) e Chakraborti e Chatterjee (2014) abordam características de texturas baseadas em *Local Binary Patterns* (LBP)

Figura 7 – Um exemplo da aplicação da transformada de Fourier em uma imagem.



Fonte: (MATERKA; STRZELECKI et al., 1998).

desde seu lançamento no trabalho de Ojala, Pietikäinen e Harwood (1996), devido ao seu bom desempenho e robustez, apesar de ser uma técnica relativamente simples como é explicado na Seção 2.3.5.1. No Livro *Computer Vision Using Local Binary Patterns* de Pietikäinen et al. (2011), é mostrada uma visão completa sobre o LBP e a aplicação dele em diversas áreas da visão computacional. O trabalho de Liu et al. (2017) também mostra uma característica importante desses experimentos: o quanto eles podem ser exaustivos baseados na quantidade de bases de dados ou descritores de texturas disponíveis para avaliar.

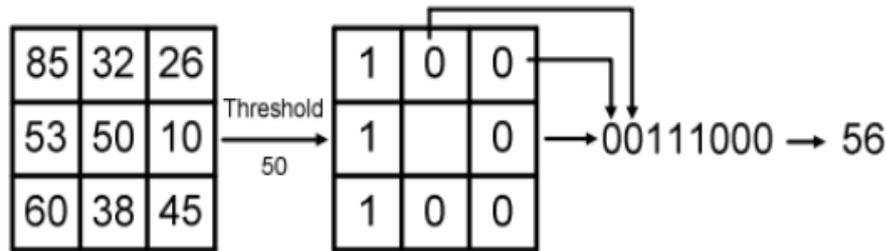
2.3.5.1 Local Binary Patterns

Ojala, Pietikäinen e Harwood (1996) introduziu o operador do *Local Binary Patterns* (LBP), derivado de uma definição geral de textura em uma vizinhança local. Ao redor de cada *pixel*, o operador pega a vizinhança local e limiariza os *pixels* da vizinhança de acordo com o *pixel* central. A resultante dessa operação é um *patch* da imagem binarizado, que é utilizado como descritor local da imagem. A definição original foi uma janela de 3×3 , resultando em um código de 8 bits baseado nos *pixels* que são vizinhos do *pixel* central. O operador do LBP é formalmente definido como

$$LBP(x_c, y_c) = \sum_{n=0}^7 2^n s(i_n - i_c) \quad (2.1)$$

onde n , nessa situação, corre sobre os vizinhos do *pixel* central c , i_c e i_n são os valores de intensidade dos pixels centrais e vizinhos respectivamente. No trabalho de Tan e Triggs (TAN; TRIGGS, 2010) é proposta uma vizinhança circular onde as localizações que não caem exatamente no centro do *pixel* são interpoladas. Um exemplo da geração do código LBP é mostrado na Figura 8.

Figura 8 – Código Gerado pelo operador LBP original.



Fonte: (OJALA; PIETIKÄINEN; HARWOOD, 1996).

Extensões adicionais do operador original do LBP foram implementadas posteriormente em (AHONEN et al., 2009) e (WANG; SHEIKH; BOVIK, 2002). Alguns padrões binários são propriedades fundamentais da textura e em tempos sua frequência excede 90%. Esses padrões são chamados de "uniformes", e são definidos por uma medida de uniformidade que correspondem a uma transição no espaço, por exemplo, 0 muda para 1 ou 1 muda para 0. Essa extensão é chamada de *Uniform LBP* (LBP_{Ru2}). A invariância a rotação é alcançada com o operador do LBP considerando o valor mínimo exclusivo dos padrões binários que é obtido deslocando a estrutura binária. Essa extensão é chamada de "Rotation invariant LBP" (LBP_{ri}).

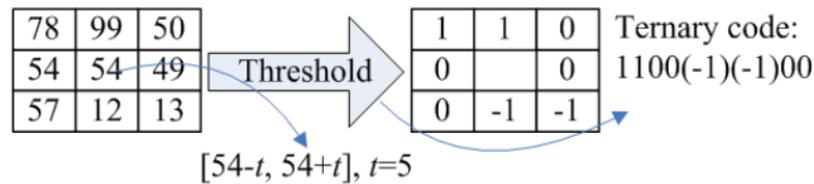
2.3.5.2 Local Ternary Patterns

Xiaoyang Tan and Bill Triggs propuseram (TAN; TRIGGS, 2010) o *Local Ternary Pattern* (LTP) em 2007 como uma generalização direta do LBP, a qual é mais discriminante e menos sensível ao ruído em regiões uniformes. Partindo da premissa que o LBP limiariza a partir do valor do *pixel* central, eles tendem a ser sensíveis ao ruído, especialmente em regiões de imagens quase uniformes como faces humanas, por exemplo. A generalização que o LTP faz do LBP é dada como

$$s'(u, i_c, t) = \begin{cases} 1, & u > i_c + t \\ 0, & |u - i_c| < t \\ -1, & u \leq i_c - t \end{cases} \quad (2.2)$$

onde ao invés de gerar um código binário (0,1) é gerando um código ternário (0,1,-1). As intensidades em uma zona em torno de i_c são quantizadas para zero, os que estão acima são quantificados para +1 e aqueles abaixo dele para -1 baseado em um limiar definido pelo usuário, gerando um código ternário. A Figura 9 mostra um exemplo de geração de um código LTP, com o limiar (t) sendo igual a cinco.

Figura 9 – Código Gerado através do operador LTP.

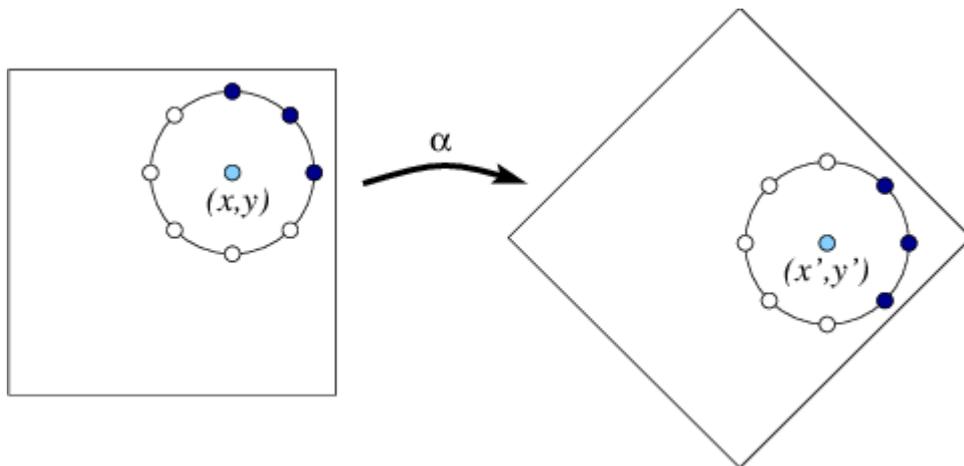


Fonte: (TAN; TRIGGS, 2010).

2.3.5.3 Local Binary Patterns Fourier Histogram

Em (AHONEN et al., 2009), foi proposto o *Local Binary Patterns Fourier Histogram* (LBP-HF) com o objetivo de gerar características invariantes a rotação. A análise de textura invariante a rotação é um problema brandamente estudado, que foca em fornecer recursos de textura que sejam invariantes ao ângulo de rotação de uma imagem de entrada. Essa característica é tipicamente robusta a mudança de iluminação. O LBP-HF utiliza o LBP uniforme ($LBP_{R_{u2}}$) combinado com uma *Discrete Fourier Transform* (DFT) para cobrir os efeitos da rotação da imagem nos pontos da vizinhança circular. A Figura 10 mostra o efeito da rotação da imagem.

Figura 10 – Efeito da rotação da imagem em uma vizinhança circular.



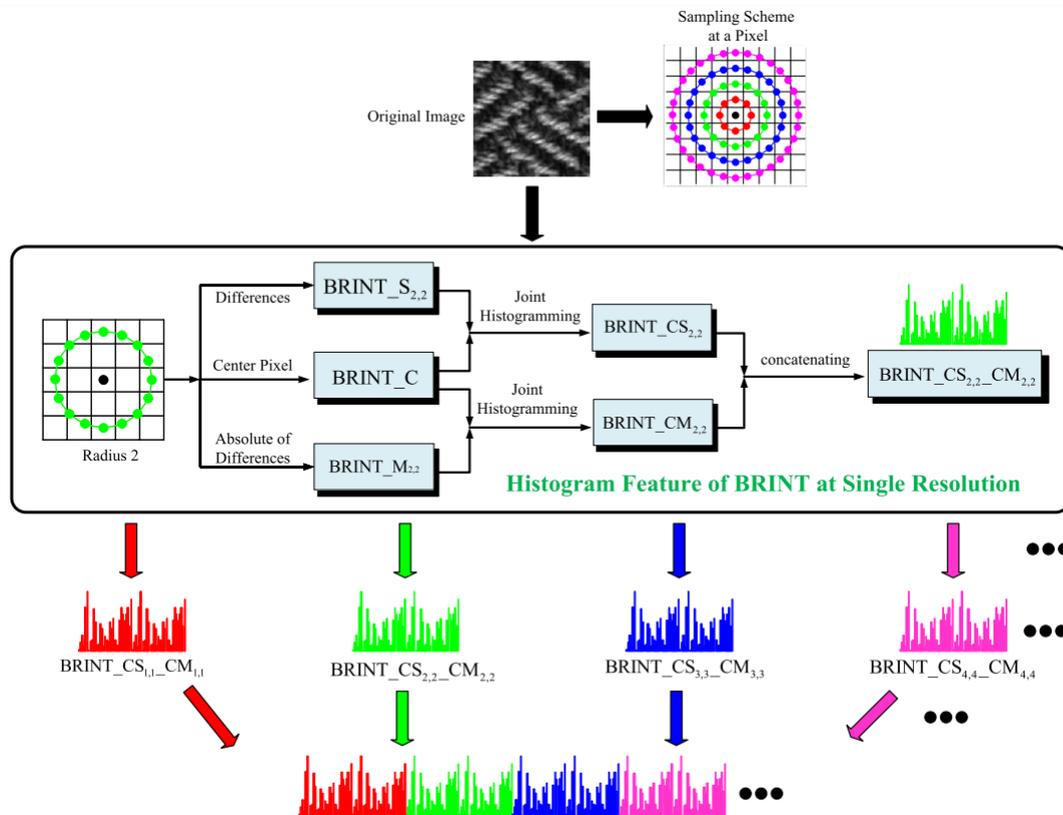
Fonte: (AHONEN et al., 2009).

2.3.5.4 Binary Rotation Invariant and Noise Tolerant

O trabalho de Liu et al. (2014) propõe o *Binary Rotation Invariant and Noise Tolerant* (BRINT), que se trata de um descritor simples, eficiente e robusto. Ele é proposto pelo fato de descritores como LBP e CLBP possuírem limitações como a falta de robustez a

ruído. O BRINT é simples pelo fato de ter uma construção rápida e compacta mantendo a eficiência esperada e robusto a problemas como variações de iluminação, mudança de rotação e ruído.

Figura 11 – Esquema do descritor BRINT.



Fonte: (LIU et al., 2014).

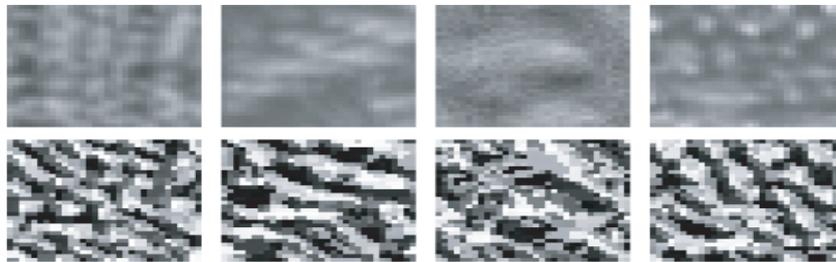
O BRINT combina três descritores individuais: BRINT_C, BRINT_M e BRINT_S e com isso ele utiliza totalmente os elementos invariantes a rotação para evitar que os padrões uniformes não tomem a proporção dominante. A Figura 11 mostra como é estruturado o histograma de características do BRINT após a combinação das três técnicas. Em sua proposta, o BRINT é comparado com técnicas como LBP(OJALA; PIETIKÄINEN; HARWOOD, 1996) LTP(TAN; TRIGGS, 2010) e CLBP(WANG; SHEIKH; BOVIK, 2002) e tem uma desempenho superior nas bases que foram avaliadas.

2.3.5.5 Binarized Statistical Image Features

Inspirado pelos métodos LBP e LPQ, o *Binarized Statistical Image Features* (BSIF) foi proposto com o trabalho de Kannala e Rahtu (2012). Para descrever a vizinhança de cada *pixel*, o BSIF calcula um código binário que é baseado na convolução da imagem de entrada com um conjunto de filtros lineares. Esses filtros lineares são aprendidos através de um conjunto de treinamento de imagens naturais, maximizando a independência estatística

das respostas desses filtros. A Figura 12 mostra as imagens de entrada e suas respectivas saídas como códigos BSIF.

Figura 12 – Imagens de entrada e suas representações através do descritor BSIF.



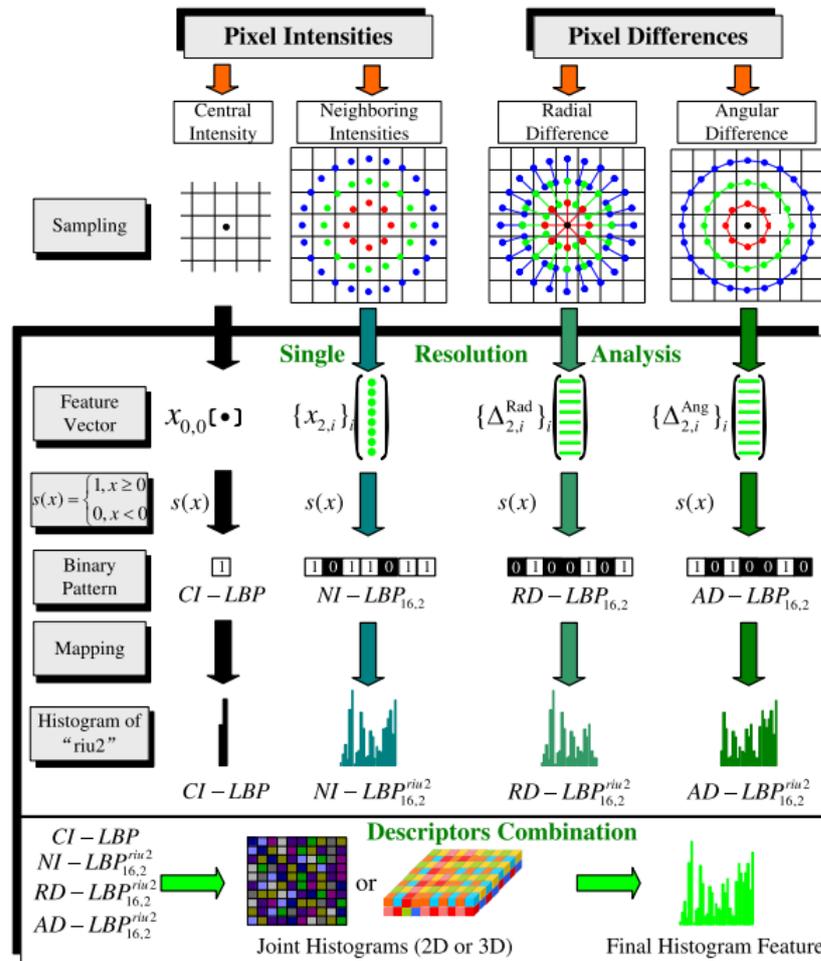
Fonte: (KANNALA; RAHTU, 2012).

Em sua proposta, o BSIF é comparado com o LBP (OJALA; PIETIKÄINEN; HARWOOD, 1996) e o LPQ (RAHTU et al., 2012) nas bases Outex (OJALA et al., 2002) e CURET (DANA et al., 1997) e no experimento mostra um desempenho superior a esses métodos. Mais tarde o trabalho de Liu et al. (2017) mostra que o BSIF tem um desempenho geral inferior a essas duas técnicas. O desempenho do BSIF nesse trabalho parece degradar quando o tamanho da vizinhança é alterado.

2.3.5.6 Central Intensity Neighbor Intensity Radial Difference

O trabalho de Liu et al. (2012) propõe o *Central Intensity Neighbor Intensity Radial Difference* (CINIRD), que é um descritor generalizado do LBP que possui duas características complementares: intensidades e diferenças de *pixels*. A característica de intensidade considera o *pixel* central e seus vizinhos dos *patches* locais, enquanto as características de diferença são calculadas através dos componentes raio e ângulo. A Figura 13 mostra um *overview* da técnica.

Figura 13 – Esquema do descritor CINIRD.



Fonte: (KANNALA; RAHTU, 2012).

A técnica é bem mais simples do que suas antecessoras, além de não ter necessidade de treinamento do dicionário de *textons*. Os experimentos de (LIU et al., 2012) utilizando esse descritor são executados nas bases Brodatz, Outex, CURET e Kth-tips2b e em sua proposta é comparado com os resultados do LBP e com o CLBP, obtendo melhores taxas de desempenho.

2.3.5.7 Completed Local Binary Count

Em Zhao, Huang e Jia (2012) é proposto o *Local Binary Count* (LBC) com o objetivo de atacar o problema de classificação de texturas incluindo variação de rotação. Ainda no mesmo trabalho também é proposto o *Completed Local Binary Count* (CLBC) motivado pelo CLBP. O trabalho também explora as diferenças entre informações micro e macroscópicas.

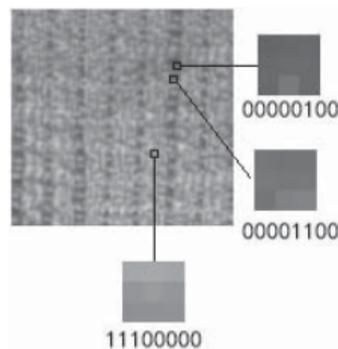
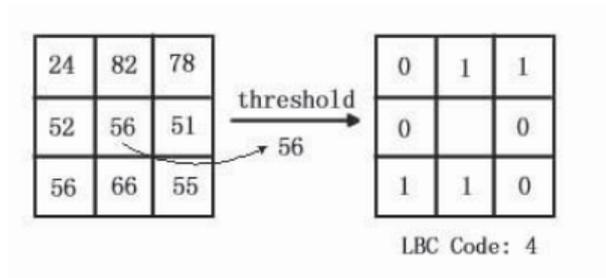
O operador LBC considera apenas os valores 1 dos vizinhos ao invés de codificá-lo. A

definição do processo de computação do LBC é definida por

$$LBC_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c), s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.3)$$

onde g_c representa os valores de cinza do pixel central e g_p denota o valor de cinza do pixel em um círculo de raio R e P é o número total de vizinhos. A Figura 14 ilustra o LBC com exemplos de textura macroscópica e microscópica. Os experimentos dessa técnica são executados nas bases CURET e Outex onde ele mostra taxas de desempenho superiores ao LBP e o CLBP.

Figura 14 – Código CLBC.

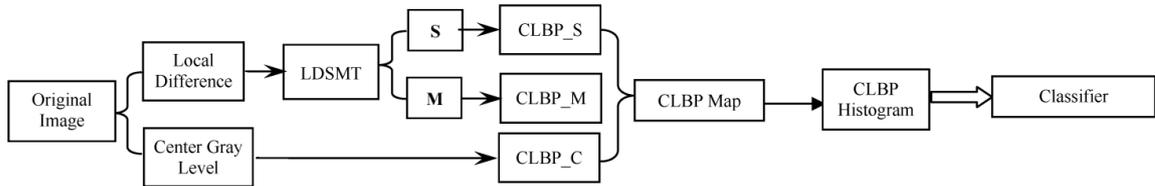


Fonte: (ZHAO; HUANG; JIA, 2012).

2.3.5.8 Completed Local Binary Pattern

No trabalho de Wang, Sheikh e Bovik (2002) é proposto o *Completed Local Binary Pattern* (CLBP) com o objetivo de representar informações que são perdidas no LBP original mas mantendo o seu estilo. No CLBP uma região local é representada pelo seu *pixel* central e por uma *Local difference sign-magnitude transform* (LDSMT). A Figura 15 mostra o *framework* do CLBP.

Figura 15 – Framework da técnica CLBP.



Fonte: (KANNALA; RAHTU, 2012).

A imagem original é primeiramente representada pelo nível de cinza do seu centro e pela diferença local. O LDSMT decompõe a diferença em componentes de sinal e magnitude. O CLBP é uma combinação de três operadores que tem o objetivo de preservar mais a informação da diferença local, sendo eles o CLBP_C, CLBP_S e O CLBP_M. Esse operador CLBP_M é definido por

$$CLBP_M_{P,R} = \sum_{p=0}^{P-1} t(m_p, c) 2^p, t(x, c) \begin{cases} 1, x \geq c \\ 0, x < c \end{cases} \quad (2.4)$$

onde c é um limiar para ser definido adaptativamente. O valor m_p é adquirido a partir de toda imagem. O CLBP_C é definido por

$$CLBP_C_{P,R} = t(g_c, cI) \quad (2.5)$$

onde t é o limiar definido na Equação 2.4 e g_c é a média de nível de cinza da imagem. Finalmente, os três operadores são concatenados para representar a imagem.

2.3.5.9 Completed Local Binary Pattern Fourier Histogram

Em Zhao et al. (2012) foi proposta uma abordagem para computar características invariantes a rotação de histogramas de padrões locais não variantes. Isso quer dizer que a estratégia é primeiro computar um histograma do tipo LBP simples em toda imagem e apenas depois construir as características invariantes a rotação a partir da DFT do histograma. Nesse trabalho, onde vários descritores locais foram combinados com a abordagem, o *Completed Local Binary Pattern Fourier Histogram* (CLBPHF) nasceu da combinação do LBPHF e do CLBP.

2.3.5.10 Discriminative Completed Local Binary Pattern

O trabalho de Guo, Zhao e Pietikäinen (2012) propõe um método de extração de características para obter padrões discriminativos. Para isso, o *framework* é formado por um modelo em camadas, que pode estimar subconjunto de padrões ótimo de uma imagem considerando sua robustez, poder discriminativo e representativo. Semelhante ao trabalho

de (ZHAO et al., 2012), essa metodologia pode ser integrada com os descritores variantes ao LBP para gerar novas características para classificação de texturas. Com isso, o *Discriminative Completed Local Binary Pattern* (disCLBP) é o resultado da combinação dessa metodologia com a técnica CLBP.

2.3.5.11 Dominant Local Binary Patterns

O *Dominant Local Binary Patterns* (DLBP) proposto por Liao, Law e Chung (2009) faz o uso dos padrões mais frequentes da imagem para capturar informação textural. O método é uma extensão do LBP convencional, com a abordagem voltada para capturar efetivamente os padrões dominantes. Com isso, o DLBP encapsula mais informações sobre a textura com a desvantagem da falta de consideração a interações distantes de *pixels*.

2.3.5.12 Extended Local Binary Patterns

O trabalho de Zhou, Wang e Wang (2008) propõe uma metodologia de extensão do LBP dado que as extensões anteriores do operador do LBP descartavam algumas informações importantes da textura. Ele combina e classifica padrões não-uniformes baseado na análise de sua estrutura e na probabilidade de ocorrência, fazendo assim com que o operador acabe usando totalmente as informações que são descartadas pelo LBP original. O operador do *Extended Local Binary Patterns* (EXLBP) é definido por

$$EXLBP_{P,R} = \begin{cases} \sum_{p=0}^{P-1} & \text{se } S(LBP_{P,R}) \leq 1 \\ \sum_{p=0}^{P-1} & \text{se } S(LBP_{P,R}) = 2, D_{ROR}^{min}(LBP_{P,R}) = 1 \\ P + 1 + B(LBP_{P,R}) & \text{se } S(LBP_{P,R}) = 2, D_{ROR}^{min}(LBP_{P,R}) = 2 \\ P + 2 + B(LBP_{P,R}) & \text{se } S(LBP_{P,R}) = 2, D_{ROR}^{min}(LBP_{P,R}) > 2 \\ P + 3 + B(LBP_{P,R}) & \text{se } S(LBP_{P,R}) > 2 \end{cases} \quad (2.6)$$

onde

$$B(LBP_{P,R}) = 3 * \left[\sum_{p=0}^{P-1} g_p / 8 \right] \quad (2.7)$$

e D_{ROR}^{min} é uma medida de similaridade. O operador do SLBP é o grupo definido pelo arranjo de zeros e 1 em uma proporção da imagem. Ele é definido por

$$S(LBP_{P,R}) = \frac{1}{2} \left(|g_{p-1} - g_0| + \sum_{p=0}^{P-1} |g_p - g_{p-1}| \right) \quad (2.8)$$

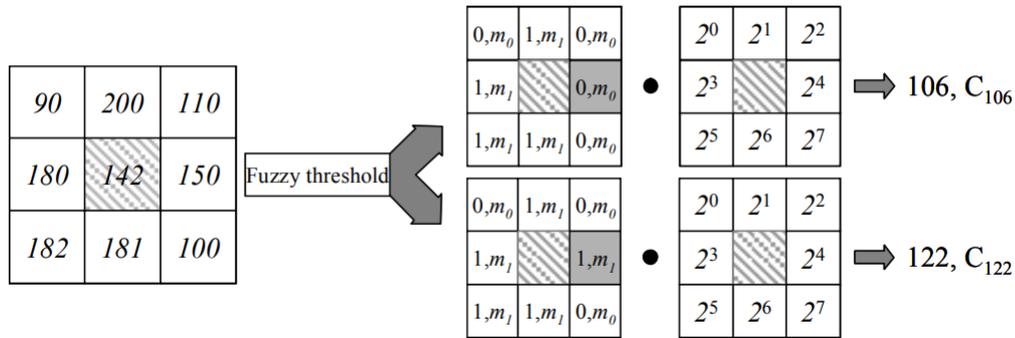
onde g_c corresponde ao valor de nível de cinza do *pixel* central e g_p aos valores de cinza de P igualmente espaçados num raio R ($R > 0$).

2.3.5.13 Fuzzy Local Binary Patterns

O trabalho de Iakovidis, Keramidas e Maroulis (2008) estende o operador original do LBP incorporando a lógica *fuzzy* para resolver o problema de caracterização de texturas de

imagens de ultrassom. Nas imagens de ultrassom existe um ruído que acaba introduzindo uma incerteza na representação de características texturais. O *Fuzzy Local Binary Patterns* (FLBP) tenta resolver essa incerteza, visto que maioria das metodologias não procuram abordar esse problema. A Figura 16 mostra o esquema de computação do FLBP.

Figura 16 – Esquema de computação FLBP.



Fonte: (IAKOVIDIS; KERAMIDAS; MAROULIS, 2008).

2.3.5.14 Improved Local Binary Patterns

Apresentado pela primeira vez no trabalho de Jin et al. (2004) como uma representação facial, o *Improved Local Binary Patterns* (ILBP) é uma melhoria do LBP original que considera tanto a forma e informação textural ao invés da informação pura de escala de cinza e sua robustez a variação de iluminação. O ILBP pode ser representado por

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - m)2^i + s(g_c - m)2^P \quad (2.9)$$

onde

$$s(x) = \begin{cases} 1, & x > 0 \\ 0, & \leq 0 \end{cases} \quad (2.10)$$

e também

$$m = \frac{1}{P+1} \left(\sum_{i=0}^{P-1} g_i + g_c \right) \quad (2.11)$$

sendo g_c a representação do nível de cinza do *pixel* central. Os experimentos mostraram que o ILBP é insensitivo a variação de iluminação.

2.3.5.15 Local Binary Pattern Difference

O *Local Binary Pattern Difference* (LBPD) foi proposto no trabalho de Hong et al. (2014) para atacar o problema derivado da limitação não numérica do LBP original. O descritor caracteriza até que ponto num LBP varia do modelo local médio de uma região de

interesse. A técnica foi formulada para manter a simplicidade do LBP original, ser de rotação invariável e computacionalmente eficiente. A diferença principal entre o LBPD e o LBP original é que o LBPD reflete a diversidade da co-ocorrência local enquanto o LBP representa isso diretamente, tornando o LBPD mais atrativo.

2.3.5.16 Local Binary Pattern Variance

O trabalho de Guo, Zhang e Zhang (2010) apresentou o *Local Binary Pattern Variance* (LBPV) como proposta para incorporar a informação de contraste local no histograma do LBP original usando a variância (VAR) como um peso adaptável para cada código LBP. Primeiro é estimado as orientações principais da textura da imagem e em seguida isso é usado para alinhar o histograma. Essa resultante é usada para medir a dissimilaridade entre as imagens. A definição de variância (VAR) é dada por

$$VAR_{P,R} = \sum_{p=0}^{P-1} (g_p - u)^2 \quad (2.12)$$

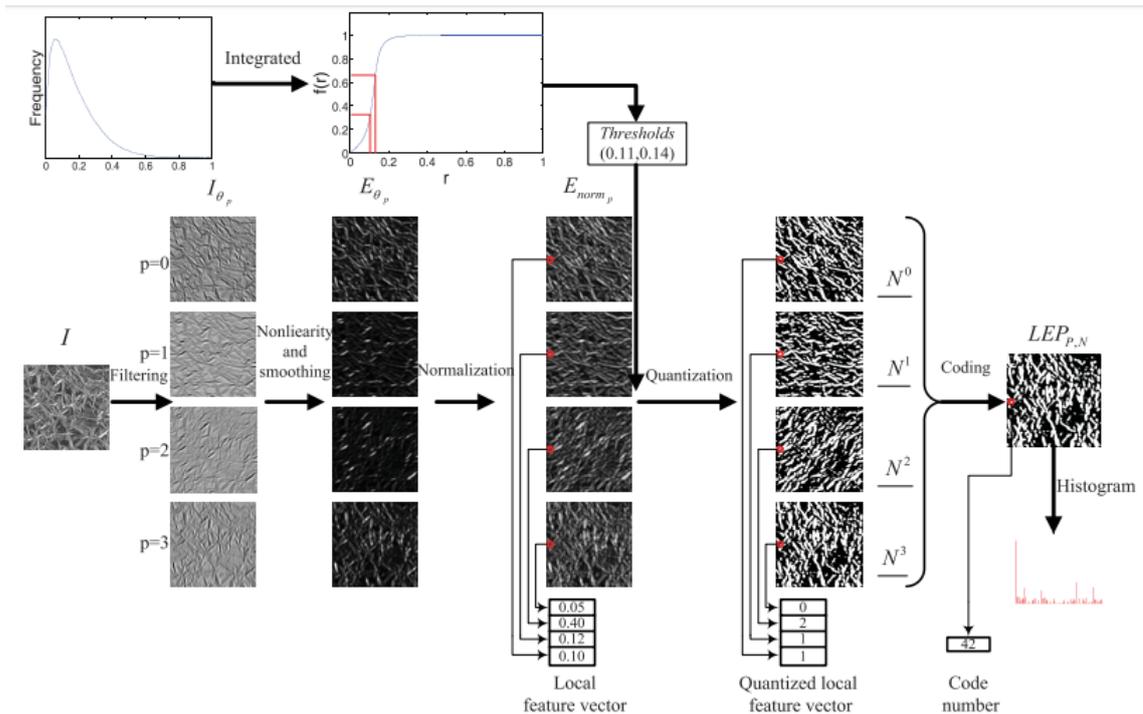
onde $u = 1/P \sum_{p=0}^{P-1} g_p$. A variância nesse caso é relatada a característica de textura. Com isso, o histograma LBPV é computada como dada a equação 2.13

$$LBPV_{P,R}(k) = \sum_{i=1}^N \sum_{j=1}^M w(LBP_{P,R}(i, j), k) \quad k \in [0, k] \quad (2.13)$$

2.3.5.17 Local Energy Pattern

O *Local Energy Pattern* (LEP) se trata de uma representação estatística baseada em histogramas, criada por Zhang, Liang e Zhao (2012). Ele utiliza energias normalizadas e orientadas localmente para gerar os vetores locais de características que descrevem as estruturas locais e em seguida cada vetor é quantizado por limiares auto-adaptativos. O LEP codifica a relação entre diferentes canais usando um código N-ário ao invés de um esquema binário. Contudo, essas mudanças propostas pelo LEP requerem um estágio extra de pré treinamento.

Figura 17 – Esquema de computação LEP.



Fonte: (ZHANG; LIANG; ZHAO, 2012).

A Figura 17 mostra o diagrama da representação de imagem do LEP. Ele possui camadas de filtragem, normalização e suavização como pré processamento antes de serem executadas as extrações dos vetores locais. O LEP se mostrou robusto a variações de múltiplas escalas e de uma mesma imagem, além de ser também ser variante a rotação.

2.3.5.18 Local Frequency Descriptor

O trabalho de Maani, Kalra e Yang (2013) propõe o *Local Frequency Descriptor* (LFD) que também é motivado pelo LBP. O descritor foi proposto para classificação de texturas, é baseado em componentes locais de frequência que são computados aplicando a transformada de *Fourier* 1D nos *pixels* de vizinhança que estão ao redor de um ponto central para cada *pixel*. Eles observam no trabalho que componentes de baixa frequência são os maiores constituintes de funções circulares que pode efetivamente representar texturas. O descritor é robusto a variância de rotação e a mudanças lineares de iluminação e ruído.

2.3.5.19 Local Quinary Patterns

O *Local Quinary Patterns* (LoQP) é proposto no trabalho de Nanni, Lumini e Brahmam (2010) em uma comparação desse método com outras variantes do LBP na resolução de problemas relacionados a aplicações de bio-imagem. O LBP original utiliza uma codificação binária para representar a diferença entre o de nível de cinza de um *pixel* central e

seus vizinhos. No LoQP a diferença entre o *pixel* central x e cada um dos seus vizinhos u pode assumir 5 valores diferentes ao invés da abordagem binária original. Essa codificação é dada por

$$d = \begin{cases} 2 & u \geq x + \tau \\ 1 & x + \tau \leq u < x + \tau_2 \\ 0 & x - \tau_1 < u < x + \tau_1 \\ -1 & x - \tau_2 \leq u < x - \tau_1 \\ -2 & \text{caso contrário} \end{cases} \quad (2.14)$$

onde t_1 e t_2 são limiares definidos previamente. A Figura 18 mostra um exemplo de código LoQP.

Figura 18 – Código LoQP.

1	-2	1
-1		2
1	-1	0

Fonte: (NANNI; LUMINI; BRAHNAM, 2010).

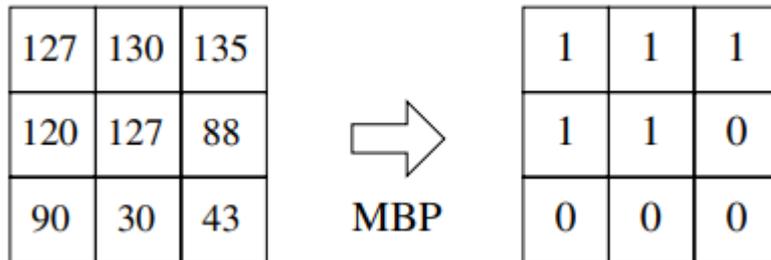
2.3.5.20 Median Binary Patterns

No trabalho de Hafiane, Palaniappan e Seetharaman (2015) o *Median Binary Patterns* (MBP) foi proposto para resolver o problema de classificação de texturas. O método se baseia em extrair estruturas locais e descrever suas distribuições por uma abordagem global. A diferença chave entre a metodologia original LBP e o MBP está na escolha do valor dos limiares. O MBP se limita a uma janela 3×3 , resultando em 512 padrões distintos. O MBP é definido por

$$MBP = \sum_{i=1}^L f(a_i) \times 2^i \quad f(a_i) = \begin{cases} 1 & \text{se } a_i \geq Med \\ 0 & \text{caso contrário} \end{cases} \quad (2.15)$$

onde L é o número de vizinhos e a_i é o valor de intensidade. A Figura 19 mostra uma janela típica de um operador MBP. Ele é robusto a mudanças monotônicas na escala de cinza já que o limiar não depende do valor de intensidade.

Figura 19 – Esquema de computação MBP.



Fonte: (HAFIANE; PALANIAPPAN; SEETHARAMAN, 2015).

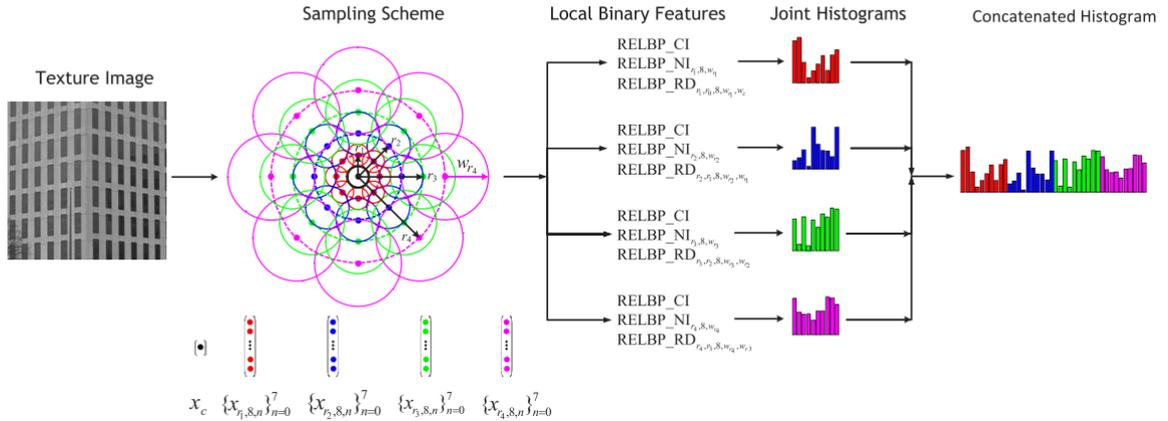
2.3.5.21 Multichannel Decoded Local Binary Patterns

O *Multichannel Decoded Local Binary Patterns* (MDLBP) foi proposto no trabalho de Schaefer e Doshi (2012) com o objetivo de atacar a perda de informação que existe quando histogramas LBP concatenados possuem diferentes raios além da ambiguidade aditiva. Ou seja, esse descritor procura codificar a informação de vários LBPs em uma mesma localização, porém com escalas diferentes. Para isso são extraídos para cada localização dos *pixels* códigos LBP em escalas diferentes enquanto a combinação desses códigos identifica o *bin* do histograma que será incrementado.

2.3.5.22 Median Robust Extended Local Binary Pattern

O *Median Robust Extended Local Binary Pattern* (MRELBP) foi proposto por Liu et al. (2016) com o objetivo de sanar uma deficiência do LBP original e algumas de suas variantes que é a sensibilidade a ruído. Essa sensibilidade os tornam incapazes de capturar informações da macroestrutura. O MRELBP compara as médias regionais da imagem ao invés da intensidade dos níveis de cinza. A Figura 20 mostra o esquema da técnica proposta.

Figura 20 – Esquema de computação MRELBP.



Fonte: (LIU et al., 2016).

2.3.5.23 Multi-scale Joint Encoding of Local Binary Patterns

O trabalho de Qi et al. (2013) propõe o *Multi-scale Joint Encoding of Local Binary Patterns* (MSJLBP) com o objetivo de cobrir a dificuldade que o MS-LBP tem ao ignorar a correlação entre diferentes escalas, perdendo muita informação discriminativa. Usando uma estratégia de junção, o MSJLBP consegue caracterizar estruturas locais da imagem mais fortes em diferentes escalas. Sendo s_1 e s_2 escalas diferentes, MSJ-LBP pode ser definido como

$$MSJ - LBP(\vec{s}_1, \vec{s}_2) = |LBP^{RIU}(\vec{s}_1), LBP^U(\vec{s}_2, i)|_{co}. \quad (2.16)$$

Essa estratégia de codificação foca exclusivamente em capturar essa relação entre diferentes escalas ao redor de um ponto. Com isso o MSJ-LBP também é invariante a escala de cinza visto que as comparações binárias são invariantes a variância da escala monotônica de cinza. Além disso, ele também é invariante a rotação e tem um custo computacional baixo.

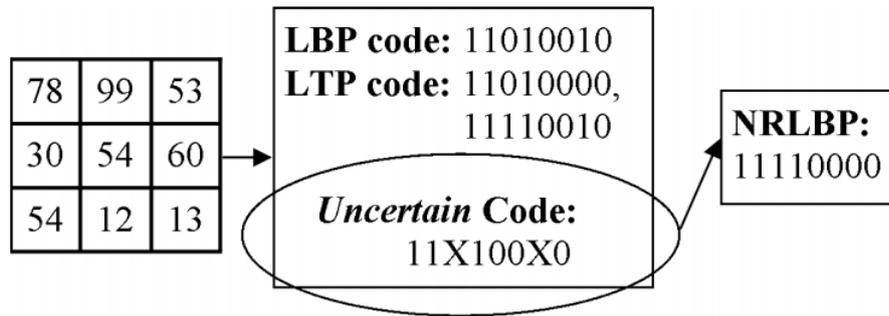
2.3.5.24 Noise-Resistant Local Binary Pattern

O *Noise-Resistant Local Binary Pattern* (NRLBP) foi proposto como mais uma alternativa a cobrir a dificuldade que o operador original do LBP tem contra ruídos no trabalho de Ren, Jiang e Yuan (2013). O NRLBP codifica a pequena diferença de *pixel* como um bit X de incerteza primeiro e depois determinar o X com base em outros bits determinados

do código LBP. Essa diferença é dada por

$$b_p^n = \begin{cases} 1 & \text{se } z_p \geq t \\ x & \text{se } |z_p| < t \\ 0 & \text{se } z_p \leq -t \end{cases} \quad (2.17)$$

Figura 21 – Código NRLBP.



Fonte: (REN; JIANG; YUAN, 2013).

onde o valor do bit de incerteza é restringido em 0 ou 1, representado por uma variável x_i . O código da incerteza é representado por

$$\overrightarrow{b_{P-1}^N b_{P-2}^N \dots b_1^N b_0^N} = C(\mathbf{X}.) \quad (2.18)$$

2.3.5.25 Noise Tolerant Local Binary Pattern

O trabalho de Fathi e Naghsh-Nilchi (2012) propõe o *Noise Tolerant Local Binary Pattern* (NTLBP) que se trata de um descritor que ataca a deficiência do LBP original em relação às imagens ruidosas. O NTLBP utiliza um filtro circular de votação majoritária além de um esquema de rotulação de rotação invariável adequado para obter padrões uniformes e não uniformes mais regulares que tenham melhor capacidade de discriminação e mais robustez contra ruído. Além disso, o descritor também é integrado com o método de filtragem média de suavização e redução de ruído. O NTLBP é definido por

$$LBP_{P,R}^{NT} = \begin{cases} \sum_{i=0}^{P-1} BC_{P,R,K}^{NT}(i) & \text{se } v(BC_{P,R,K}^{NT})4 \end{cases} \quad (2.19)$$

onde

$$BC_{P,R,K}^{NT}(i) = CMaj_{2k+1}^i(BC_{P,R}) \quad (2.20)$$

e

$$CMaj_{2k+1}^i(BC_{P,R}) = \left\lceil \left[\sum_{j=-k}^k BC_{P,R}((P+i+j) \bmod(P)) \right] / (k+1) \right\rceil \quad (2.21)$$

onde o filtro de votação por maioria circular (CM_{aj2k1}) retorna o valor da maioria de $2k + 1$ bits adjacentes em torno do bit i no código binário de LBP (BCP, R), k -bits em cada lado. A introdução de um filtro de votação majoritária circular tem o objetivo de remover o ruído produzindo padrões regulares.

2.3.5.26 Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern

Desenvolvido para alcançar a invariância global de rotação, o *Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern* (PRICoLBP) foi proposto no trabalho de Qi et al. (2014). Para isso, ele faz uma amostragem do ponto vizinho ao longo da direção do gradiente da unidade ou da direção normal da unidade no ponto central. Diferentemente das abordagens até o momento de seu trabalho, o PRICoLBP não só pode capturar a informação de co-ocorrência do contexto espacial de forma eficaz, mas também possui invariância de rotação. O PRICoLBP é definido pela equação

$$PRICoLBP(A, B) = [LBP^{ru}(A), LBP^u(B, i(A))]_{co}, \quad (2.22)$$

onde $i(A)$ que pertence a $0, n-1$ é um índice, que pode ser determinado pelo ponto A . Sendo assim, sendo dois pontos A e B o PRICoLBP codifica a característica no ponto A com LBP_{ri} , e então codifica a característica no ponto B com LBP uniforme com referência ao ponto A . O trabalho de Qi et al. (2014) também mostra uma série de aplicações em potencial do descritor, desde reconhecimento de alimentos até classificação de cenas.

2.3.5.27 Rotation Invariant Local Phase Quantization

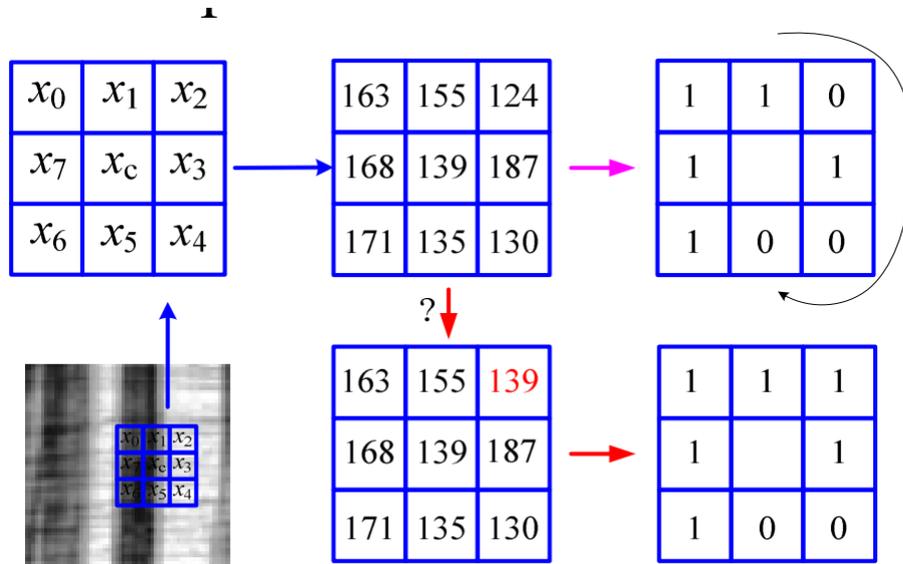
O trabalho de Rahtu et al. (2012) propôs o *Rotation Invariant Local Phase Quantization* (RILPQ) como a versão melhorada do *Local Phase Quantization* (LPQ). O método melhorado consiste em dois passos: uma estimativa da orientação das características locais e a extração do vetor binário. Sendo insensíveis a imagens desfocadas, a fase do coeficiente da transformada de *Fourier* é aplicada em ambas as etapas. Ao contrário da maioria dos descritores, ele mantém apenas as informações de fase da imagem. Com isso, o método também é robusto a variação de iluminação.

2.3.5.28 Robust Local Binary Pattern

Assim como alguns descritores, o *Robust Local Binary Pattern* (RLBP) é um descritor baseado diretamente na forma original do LBP, com aplicações em diversos campos da

análise de texturas (CHEN et al., 2013). Similarmente a outros descritores, ele foi proposto como uma melhoria baseada na dificuldade do LBP com ruídos de imagem. No RLBP é feita uma modificação na codificação, que é mostrada na Figura 22.

Figura 22 – Código RLBP.



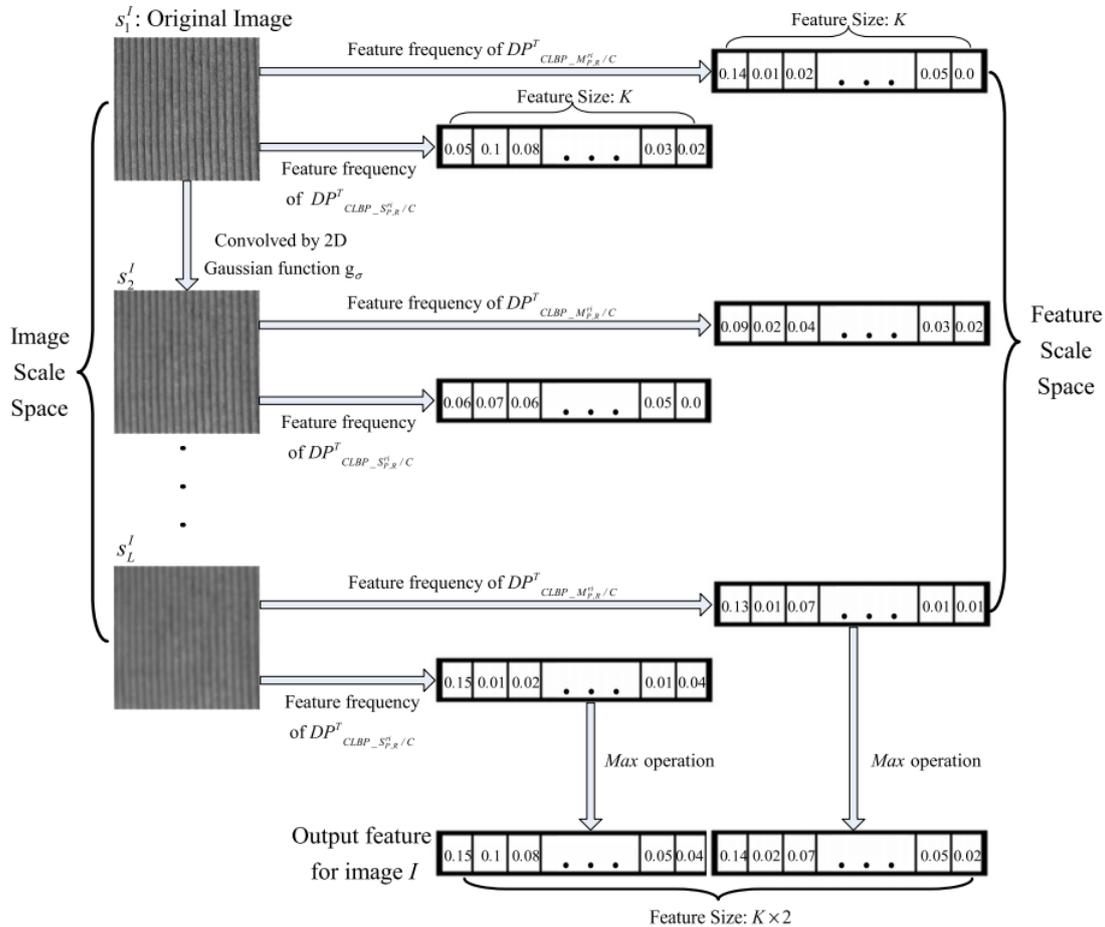
Fonte: (CHEN et al., 2013).

Os vizinhos do *pixel* central dariam um código LBP 11010011 e, seguindo o valor do *pixel* $x_2 = 124$ em comparação com toda sua vizinhança denota uma probabilidade alta de ser um ponto de ruído. Quando é feita uma alteração nesse *pixel* baseado no *pixel* central, o código LBP (que passará a ser 11110011) mostra um canto local, que é um padrão mais forte para representação.

2.3.5.29 Scale Selective Local Binary Patterns

O trabalho de Guo et al. (2015) propôs o *Scale Selective Local Binary Patterns* (SSLBP), atacando a dificuldade do LBP em conseguir bons resultados em classificação de texturas que possuam transformações de escala. A metodologia utiliza o CLBP para extrair características primeiro após aplicar um filtro gaussiano e depois aplica um esquema seletivo de escala para calcular o SSLBP. O histograma finalmente é gerado para cada imagem no espaço de escalas. A Figura 23 mostra o processo de geração do código SSLBP.

Figura 23 – Código SSLBP.



Fonte: (GUO et al., 2015).

2.3.6 Representação de texturas com Redes Convolucionais

A partir do trabalho de (ALEX; ILYA; HG, 2012), tornou-se evidente que as *Convolutional Neural Networks* (CNN) não só podem ser usadas como representações de textura como também fazem esse trabalho de maneira muito eficiente e com isso houve um acréscimo considerável de abordagens de representação de texturas usando redes convolucionais. Isso se deu pelo fato de que nesse estudo ocorreu uma taxa de classificação de imagens nunca vista até o ano de 2012.

2.3.6.1 ResNet

A *Residual Network* (ResNet) (HE et al., 2016) foi introduzida por pesquisadores da Microsoft em 2016. A rede venceu o ILSVRC 2015 ficando com uma média *top-5* de erro de 3.6, o que supera a taxa de classificação alcançada por humanos os quais obtêm *top-5* de erro entre 5 – 10.

Podendo assumir diversas configurações de camadas como mostra a Figura 24, a

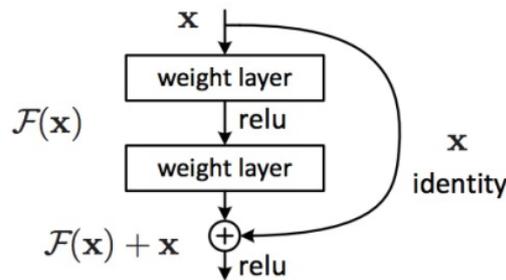
Figura 24 – Possíveis arquiteturas da ResNet.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fonte: (ALEX; ILYA; HG, 2012).

ResNet possui a ideia de blocos residuais, onde uma entrada x passa por operações de convolução-relu-convolução. A resultante disso é aplicada a entrada original como mostra a Figura 25.

Figura 25 – Resíduo da ResNet.



Fonte: (ALEX; ILYA; HG, 2012)

Contudo, também para se treinar uma rede profunda é necessária uma quantidade muito grande de dados, impondo uma limitação para problemas que não possuam uma base com amostras suficientes. A literatura mostra que tanto modelos de CNN's pré-treinados quanto convolucionais profundos podem ser aplicados para representação de textura, contudo, o tempo de extração e uso da rede neural é extremamente elevado. O trabalho de Liu et al. (2017) mostra, por exemplo, bons resultados da ScatNet e da PCA-Net, mas também mostra que o tempo que essas técnicas levam para extrair características

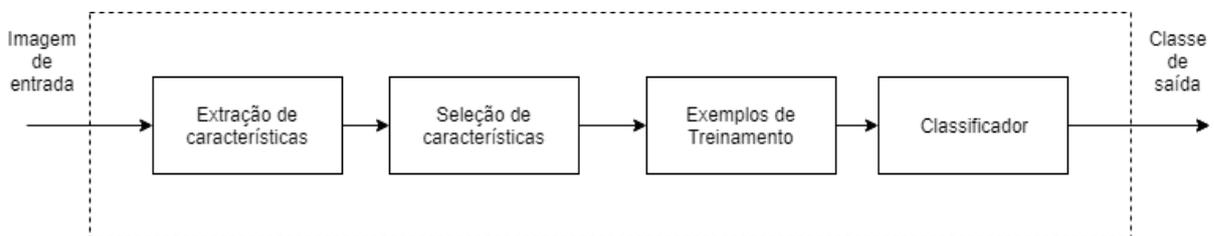
é naturalmente elevado. Dependendo da situação é possível que seja mais vantajoso o uso de descritores locais visto que não existe tanto ganho de desempenho geral.

2.4 CLASSIFICAÇÃO DE TEXTURAS

Temos por classificação a ação ou tarefa de atribuirmos a um objeto, ou incidência do mesmo, a uma determinada classe ou categoria que é conhecida previamente (ARMI; ERSHAD, 2019). Dentro da análise de texturas, a classificação é um dos quatro problemas relacionados ao tratamento de texturas em processamento de imagens. Como um problema clássico de reconhecimento de padrões, a classificação de texturas consiste principalmente de dois subproblemas críticos: representação de textura e classificação (JAIN; DUIN; MAO, 2000).

Na tarefa de classificação de texturas, um conjunto de dados de treino é usado para treinar um modelo de forma supervisionada, e esse modelo treinado é usado para definir a categoria de textura a que uma imagem de teste observada pertence. Para conseguir isso, é necessário ter um conhecimento *a priori* das classes a serem reconhecidas. Uma vez que este conhecimento esteja disponível e os recursos de textura sejam extraídos, em seguida, usam-se técnicas clássicas de classificação (TUCERYAN; JAIN, 1993). O modelo clássico de classificação é mostrado na Figura 26.

Figura 26 – Framework clássico da classificação de texturas.



Fonte: Elaborado pelo autor, 2019.

Uma regra de decisão é então aprendida para rotular uma imagem de teste desconhecida para uma determinada classe com base em sua projeção no espaço formulado pelo modelo. Várias medidas de distância e classificadores são usados na classificação de texturas. As medidas estimam a similaridade dos descritores no espaço de características (por exemplo, distância euclidiana e qui-quadrado). Os classificadores mais comuns utilizados são o KNN e o *Support Vector Machine* (SVM). Uma medida de distância é normalmente usada em um classificador que atribui uma imagem de teste ao voto principal das amostras de treinamento mais próximas no espaço de recurso.

A maioria dos conjuntos de dados de classificação de textura propõe a avaliação de texturas de materiais, como as bases ALOT (BURGHOUTS; GEUSEBROEK, 2009), Outex (OJALA et al., 2002), kth-tips-2b (MALLIKARJUNA et al., 2006), CURET (DANA et al., 1997), UIUC (LAZEBNIK; PONCE, 2005) e UMD (XU; JI; FERMÜLLER, 2009). Estes os conjuntos

de dados contêm imagens de textura com rótulos de classe. É uma prática comum dividir um conjunto de dados em conjuntos de treinamento e teste para avaliar o desempenho de um algoritmo de classificação. O desempenho de um algoritmo de classificação de textura é tipicamente avaliado através de métricas como precisão, média e desvio padrão em bases como as citadas acima. Uma visão mais detalhada sobre as bases de dados é vista na Seção 2.5, no final deste capítulo.

2.4.1 K-Nearest Neighbor

O *K-Nearest Neighbor* (KNN) é bastante usado em problemas de classificação. Ele toma medidas de distâncias, ou semelhanças, entre os pares de dados para executar a classificação. Em geral, quando se trata de valores numéricos, medidas de distâncias são usadas. A mais comum delas é a distância euclidiana, dada pela equação 2.23.

$$DistanciaEuclidiana(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.23)$$

KNN armazena os exemplos de treinamento e seus respectivos rótulos de classes x_i, c_i . O objetivo é classificar os exemplos de teste que tem o rótulo desconhecido, então dada uma instância x se deseja obter o seu rótulo c . A determinação do rótulo é feita computando a distância entre o exemplo de teste e todos os exemplos de treinamento. Então são selecionados os k vizinhos mais próximos de x , e o rótulo atribuído ao teste será o mais comum entre esses vizinhos como mostram o conjunto de equações

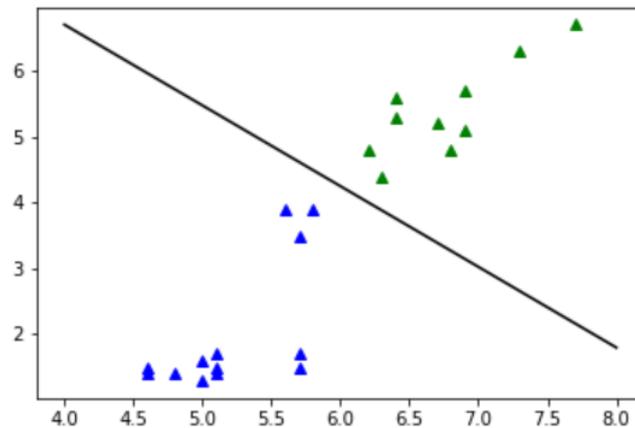
$$\begin{cases} c(x) = \arg \max_{c \in C} \sum_{i=1}^k \delta(c, c(z_i)) \\ \delta(c, c(z_i)) = 1 \text{ se } c = c(z_i) \\ \delta(c, c(z_i)) = 0 \text{ caso contrário,} \end{cases} \quad (2.24)$$

onde z_1, z_2, \dots, z_k são os k vizinhos mais próximos de x e C o conjunto de classes do problema. É esperado que classes semelhantes sejam projetadas com uma pequena distância no espaço. A distância pode ser determinada pela Distância Euclidiana citada anteriormente, entre as classes e a classificação via 1-NN, KNN com $k=1$.

2.4.2 Support Vector Machine

Sendo uma técnica bastante explorada para problemas de classificação binária, o *Support Vector Machine* (SVM) é uma técnica de aprendizagem de máquina que tem por objetivo encontrar um hiperplano em um espaço N-dimensional que possa classificar os dados (GHOLAMI; FAKHARI, 2017). Desenvolvida no final dos anos 90, baseada na teoria de Vapnik–Chervonenkis dos anos 60, foi proposta como uma solução não-linear para tarefas de classificação e regressão (CORTES; VAPNIK, 1995). A Figura 27 mostra um hiperplano construído em um problema de dados linearmente separáveis.

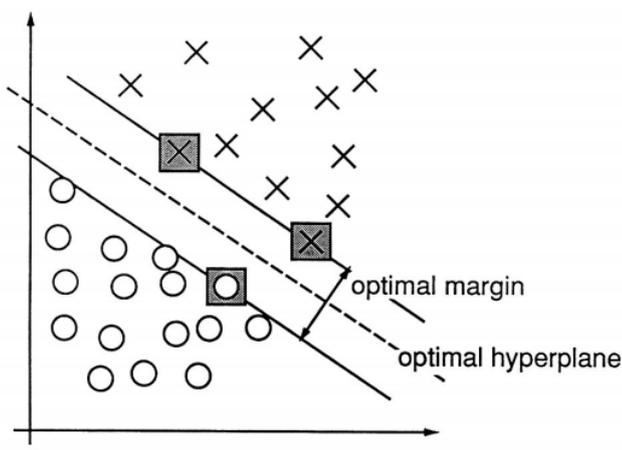
Figura 27 – Problema linearmente separável e hiperplano de separação.



Fonte: Elaborado pelo autor, 2019.

A separação de duas classes pode ser definida por diversos hiperplanos. O SVM tenta encontrar um plano ótimo, tentando maximizar a margem de ambos os pontos de entrada de cada classe. Isso reforça o poder de classificação para que novos dados possam ser classificados com uma maior certeza (GHOLAMI; FAKHARI, 2017). Esses hiperplanos são considerados limites de decisão, que são a base para a classificação do conjunto de dados. Dependendo do número de entradas, o hiperplano pode ter N dimensões. A Figura 28 mostra o hiperplano ótimo junto a vetores de suporte.

Figura 28 – Hiperplano ótimo e vetores de suporte.



Fonte: (CORTES; VAPNIK, 1995).

Por fim, os vetores de suporte são os dados que estão mais próximos do hiperplano, que influenciam na orientação e posição do mesmo. Qualquer alteração nesses vetores de suporte causa uma mudança na posição do hiperplano, já que são eles que ajudam a construir o SVM.

2.5 BASES DE TEXTURA

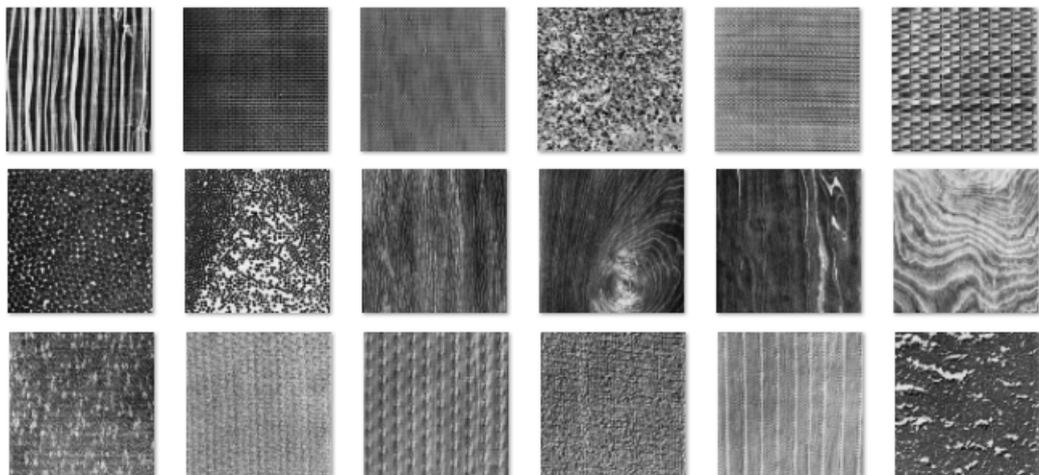
Análise de textura tem várias aplicações em diferentes áreas de visão computacional, e devido a isso, vários conjuntos de dados ficaram famosos por seu extenso uso e fácil acesso. Humanos implicitamente dependem de propriedades dos materiais que compõem objetos comuns para guiar nossas interações (SCHWARTZ; NISHINO, 2019).

Embora existam vários conjuntos de dados de textura, a maioria deles é baseada em vários materiais e objetos, embora parcialmente eles também cubram texturas naturais e cenas. Os conjuntos de dados tem um dos papéis principais no progresso do campo. Eles não apenas alinham a comparação do desempenho de algoritmos similares concorrentes como também evidenciam problemas cada vez mais complicados e desafiadores no campo.

2.5.1 Brodatz

A Brodatz (HERSEY; BRODATZ, 1968) é a base de texturas mais famosa, derivada do álbum de mesmo nome (BRODATZ, 1966). Ela possui 112 classes, porém apenas um exemplo por classe. O álbum Brodatz contém 112 imagens com tamanho 640×640 e 256 valores de cinza após a digitalização, mostrando uma variedade de texturas, pequenas e grandes, que foram coletadas para fins artísticos. As imagens da Brodatz são bastante consideradas também pelo fato delas terem sido capturadas com controle de iluminação, sendo assim, gerando imagens com uma qualidade alta. A Figura 29 mostra exemplos de algumas classes de texturas da Brodatz.

Figura 29 – Exemplos de classes de textura da Brodatz.



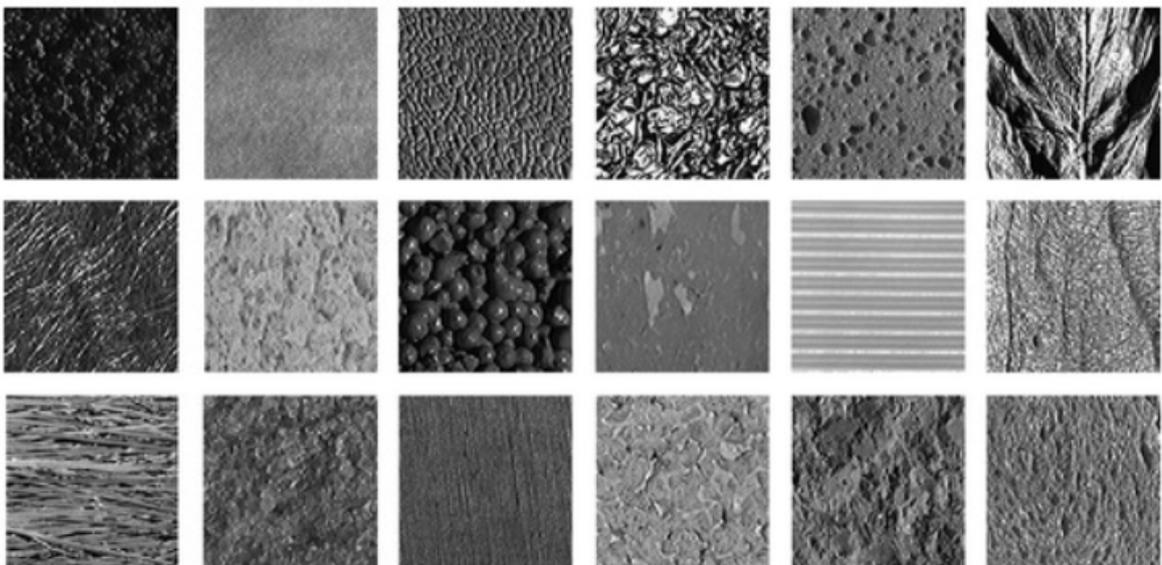
Fonte: Brodatz (1966).

Em muitas das avaliações de algoritmos, o banco de dados inteiro da base não é empregado. Em (OJALA; PIETIKAINEN, 1998), por exemplo, é utilizado apenas 15 texturas da base Brodatz em seu protocolo.

2.5.2 CURET

Com a evolução do reconhecimento de texturas, que passou a se preocupar em classificar texturas do mundo real com grandes variações intraclasses (mudanças na iluminação, pose e rotação), vários conjuntos de dados surgiram baseados em várias instâncias do mundo real. A CURET (DANA et al., 1999) foi desenvolvida pela *Columbia University and Utrecht University* e está entre um dos mais famosos conjuntos, que é considerada uma melhoria da base Brodatz. Possui blocos com diferentes categorias de defeitos, incluindo danos físicos, furos de pinos, imperfeições texturais, erros de registro de padrões, entre outros. Os exemplos de teste têm uma resolução de 512×512 pixels com 61 classes de materiais diferentes. Entretanto, a CURET tem limitações de nenhuma mudança significativa de escala para a maioria das texturas e rotação limitada no plano. A Figura 30 mostra exemplos de classes de textura da base CURET.

Figura 30 – Exemplos de classes de textura da CURET.

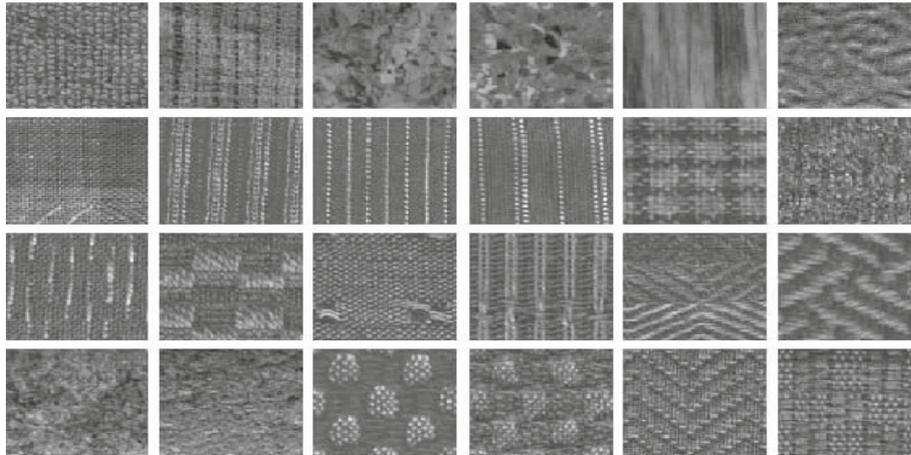


Fonte: (DANA et al., 1999).

2.5.3 Outex

A base de dados Outex foi produzida pela da Universidade de Oulu (OJALA et al., 2002). Esse conjunto de dados de imagens contém uma grande coleção de texturas, tanto na forma de texturas de superfície quanto em cenas naturais. Se trata da maior variação de classes de textura em uma única base de dados (320), com cada classe tendo imagens com variações de três iluminações diferentes e nove ângulos de rotação, com variações de escala limitadas. A Outex é um modelo para avaliação empírica de algoritmos de classificação de textura e segmentação.

Figura 31 – Exemplos de classes de textura da Outex.



Fonte: (OJALA et al., 2002).

2.5.4 KTH

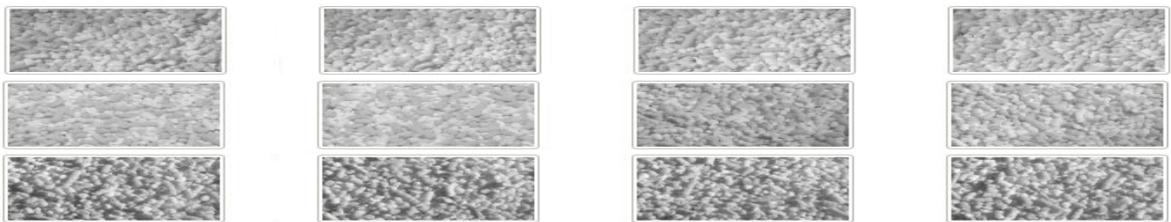
Em (HAYMAN et al., 2004) e (MALLIKARJUNA et al., 2006) pesquisadores do *Royal Institute of Technology* introduziram um conjunto de dados chamado “Texturas KTH sob diferentes iluminações, pose e escala” (KTH-TIPS), que se trata de uma expansão da CURET fotografando novas amostras de dez das classes de textura da CURET em um subconjunto dos ângulos de visualização e iluminação usados anteriormente na CURET, mas também em várias escalas. Cada classe finaliza contabilizando 81 imagens, sendo imagens de 200×200 .

Posteriormente em (CAPUTO; HAYMAN; MALLIKARJUNA, 2005) é mostrada a KTH-TIPS2, que fornece uma extensão da KTH-TIPS original imaginando várias amostras diferentes de diversos materiais. Muitos desses materiais têm estruturas 3D, o que implica que sua aparência pode mudar consideravelmente quando elementos como pose e iluminação são alterados adicionando uma complexidade ainda mais elevada do que as bases anteriores a ela. Na Figura 32 são mostrados alguns exemplos das bases KTH-Tips e KTH-Tips2.

2.5.6 ALOT

Amsterdam Library of Textures (ALOT) é uma base de texturas criada pela Universidade de Amsterdam (BURGHOUTS; GEUSEBROEK, 2009), que possui 250 classes de texturas capturadas sobre um controle de ambiente com 8 variações de iluminação distintas como mostra a Figura 34. Apesar da grande quantidade de classes bem acima da média quando se compara com UMD, possui uma complexidade muito baixa em relação a ângulo de captura, rotação e iluminação se tornando uma base não tão desafiadora.

Figura 34 – Exemplos de classes de textura da base de dados ALOT.



Fonte: (BURGHOUTS; GEUSEBROEK, 2009).

2.6 DISCUSSÃO

Neste capítulo foi mostrado o estado da arte da análise de texturas. Foi mostrado o modelo clássico de extração de características e todas as abordagens clássicas de extração. Também foi sumarizado os descritores de locais de texturas que foram propostos durante os anos que foram influenciados pelo LBP. Percebe-se que esses descritores tentam atacar problemas diversos, possuindo diferentes características em relação ao processamento de informações texturais. A Tabela 1 mostra todos os descritores que foram mostrados, bem como suas diferenças em relação à robustez, configurações de execução e desempenho no tempo de extração de características.

Tabela 1 – Sumário das técnicas de descrição local utilizadas no estudo. Ruído, Rotação e Iluminação mostram se o descritor é robusto ao determinado problema. "Parcial" na coluna de ruído quer dizer que o descritor é robusto apenas aos ruídos de desfoque e ruído gaussiano.

Método	Esquema	Operador	Tempo de extração	Dimensões	Ruído	Rotação	Iluminação
LBP	riu2	19x19	Muito Rápido	210	Não	Sim	Sim
LBP	ri	7x7	Muito Rápido	108	Não	Sim	Sim
ILBP	riu2	19x19	Muito Rápido	420	Não	Sim	Sim
MBP	riu2	19x19	Rápido	420	Não	Sim	Não
LTP	riu2	19x19	Rápido	420	Não	Sim	Não
LoQP	riu2	19x19	Rápido	840	Não	Sim	Não
FLBP	riu2	11x11	Muito Lento	50	Não	Sim	Não
NRLBP	riu2	11x11	Rápido	50	Não	Sim	Não
RLBP	riu2	19x19	Moderado	210	Não	Sim	Sim
NELBP	Padrão	19x19	Muito Rápido	273	Não	Sim	Sim
NTLBP	Padrão	17x17	Rápido	388	Não	Sim	Sim
MDLBP	riu2	11x11	Muito Rápido	1000	Não	Sim	Sim
PRICoLBP	Padrão	13x13	Rápido	3540	Não	relativo	Sim
MSJLBP	Padrão	7x7	Moderado	3540	Não	relativo	Sim
DLBP	Padrão	19x19	Moderado	14.150	Não	Sim	Sim
disCLBP	Padrão	11x11	Moderado	7796	Não	Sim	Sim
LEP	ri	32x32	Rápido	520	Parcial	Não	Não
CLBP	riu2	9x9	Rápido	3552	Não	Sim	Sim
CINIRD	riu2	7x7	Rápido	2200	Não	Sim	Sim
BRINT	ri	19x19	Rápido	1296	Parcial	Sim	Sim
MRELBP	riu2	17x17	Rápido	800	Sim	Sim	Sim
LBP	riu2	19x19	Moderado	3360	Não	Sim	Sim
LBPV	riu2	15x15	Moderado	158	Não	Sim	Sim
CLBPHF	u2	19x19	Rápido	4580	Parcial	Sim	Sim
LBDP	Padrão	7x7	Rápido	289	Parcial	Sim	Sim
SSLBP	Padrão	19x19	Rápido	2400	Não	Sim	Sim
RILPQ	Padrão	13x13	Rápido	256	Parcial	Sim	Sim
BSIF	lbp	5x5	Muito Rápido	1024	Parcial	Não	Sim
LFD	N/a	19x19	Rapido	264	Parcial	Sim	Sim

Fonte: Elaborado pelo autor, 2019.

Foram categorizadas algumas das bases de texturas que são utilizadas na literatura de análise de texturas, mostrando exemplos, apontando diferenças e expondo os desafios que as mesmas possuem. Nota-se que as bases de dados possuem características peculiares em relação ao ambiente em que as imagens foram capturas bem como os problemas que elas propõem em reportar.

Também é notório que não existe uma padronização na organização dos dados nem uma preocupação com quantidade de arquivos entre outros detalhes. A Tabela 2 mostra um sumário sobre os pontos discutidos em relação às bases usadas no estudo.

Tabela 2 – Diferenças entre as bases de dados.

	Base de textura	Ano	# de Imagens	# de Classes	Tamanho da imagem	Ambiente	Mudança de escala	Mudança de rotação	Mudança de iluminação	Perspectiva
1	Brodatz	1966	112	112	640x640	controlado	não	não	não	não
2	CUReT	1999	5612	92	200x200	controlado	sim	sim	sim	não
3	Outex_TC10	2002	4320	320	746x538	controlado	não	sim	sim	não
4	UIUC	2005	1000	25	640x480	controlado	sim	sim	sim	sim
5	KTH-TIPS2b	2006	4752	11	200x200	controlado	sim	sim	sim	sim
6	UMD	2009	1000	25	1280x960	livre	sim	sim	sim	sim
7	ALOT	2009	25000	250	1536x1024	controlado	não	não	sim	não

Fonte: Elaborado pelo autor, 2019.

3 MÉTODOS PROPOSTOS

3.1 MOTIVAÇÃO

Dada a variação das categorias de textura que existem, encontrar um descritor ou um conjunto de técnicas que seja eficiente para cada uma delas é um trabalho exaustivo. Como citado na Seção 2.5 existem diversas bases de dados que são utilizadas na classificação de texturas e esse número cresce com o passar dos anos e da evolução tecnológica.

Sempre que se tem um novo conjunto de dados, é necessário aplicar todo o modelo de classificação para sabermos quais técnicas são eficientes para representar essas texturas presentes na base. Independentemente do poder computacional que tem acesso, ainda assim é um trabalho que demanda tempo, capacidade de armazenamento e manipulação de dados, entre outros aspectos inerentes da tarefa de análise de texturas. O trabalho de Liu et al. (2017) é um exemplo, onde são processadas diversas bases de dados em 32 descritores. Considerando um contexto macro de otimização, o trabalho considera apenas os descritores que são derivados da engenharia de características, visto que métodos baseados em CNN vão possuir dados elevados de complexidade computacional, tempo de extração e custo. Mesmo para um exemplo direcionado dessa forma, é difícil calcular quanto tempo foi gasto para fazer a análise desses descritores em um número modesto de bases de texturas quando consideramos a quantidade de bases e descritores que existem.

O objetivo desse trabalho é propor um *framework* que facilite a solução desse problema. Dada uma base de dados nova, o modelo indica quais técnicas funcionam melhor para fazer a classificação de texturas baseado na similaridade dessa base com outra previamente conhecida. O modelo gera um espaço textural baseado em todas as imagens que estão inclusas nas bases de dados, e quando essa base de dados é exposta a esse espaço, sabemos onde a mesma é localizada e assim atribuímos a nova base ao espaço que ela é pertencente. Com isso, se pode ganhar todo o tempo que teria que ser dedicado para executar todos os descritores para uma determinada base desconhecida já sabendo quais técnicas melhor funcionam nela; por exemplo, em uma coleção de 32 descritores, com o resultado do *framework* podemos reduzir essa execução para 5 descritores. A Seção 3.2 detalha cada um dos componentes desse modelo.

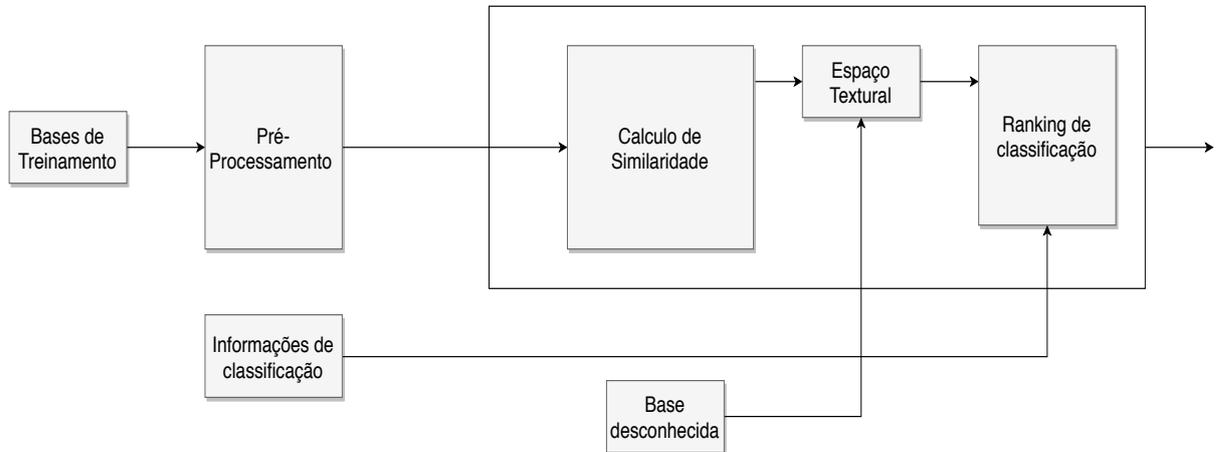
3.2 ARQUITETURA PROPOSTA

Dado este cenário, o modelo gerará um espaço de texturas baseado na similaridade entre as bases de dados, partindo da hipótese de que bases similares vão ter técnicas de classificação e descrição idênticas ou similares como as técnicas que melhor as descrevem. A Figura 35 mostra cada uma das camadas detalhadas nessa Seção.

As bases de texturas coletadas que são disponibilizadas na comunidade científica, quase sempre, não obedecem a nenhum padrão de formatação ou nomeação de seus dados visto que dado padrão de formatação não existe. A Tabela 2 da Seção 2.6 mostra as diferenças que as bases podem ter apesar de estarem muitas vezes tratando do mesmo problema. Com isso, bases diferentes possuem formatos e aspectos distintos, entre outras características. O bloco de pré-processamento trata de analisar todas as bases que serão utilizadas no treinamento de modo a normalizar todos os dados para que fiquem semelhantes: dependendo do conjunto de bases que será utilizado, podem ser aplicadas técnicas de redimensionamento ou recorte de tamanhos, aplicação de filtros, entre outros. Isso também é importante para execução geral da rede neural, que tem a limitação de aceitar as imagens de entrada somente se elas possuírem as mesmas medidas de largura e altura.

Em seguida, todos os exemplos de imagens são colocados como entrada da rede profunda *Triplet Network* com uma função de perda *Triplet Loss*. Essa rede profunda e sua função são discutidas com mais detalhes na Seção 3.2.1.2. Ela é responsável pela geração do espaço textural que divide as bases de treinamento em regiões que definem que o quão mais distante uma base n esteja de outra, menos semelhante ela é.

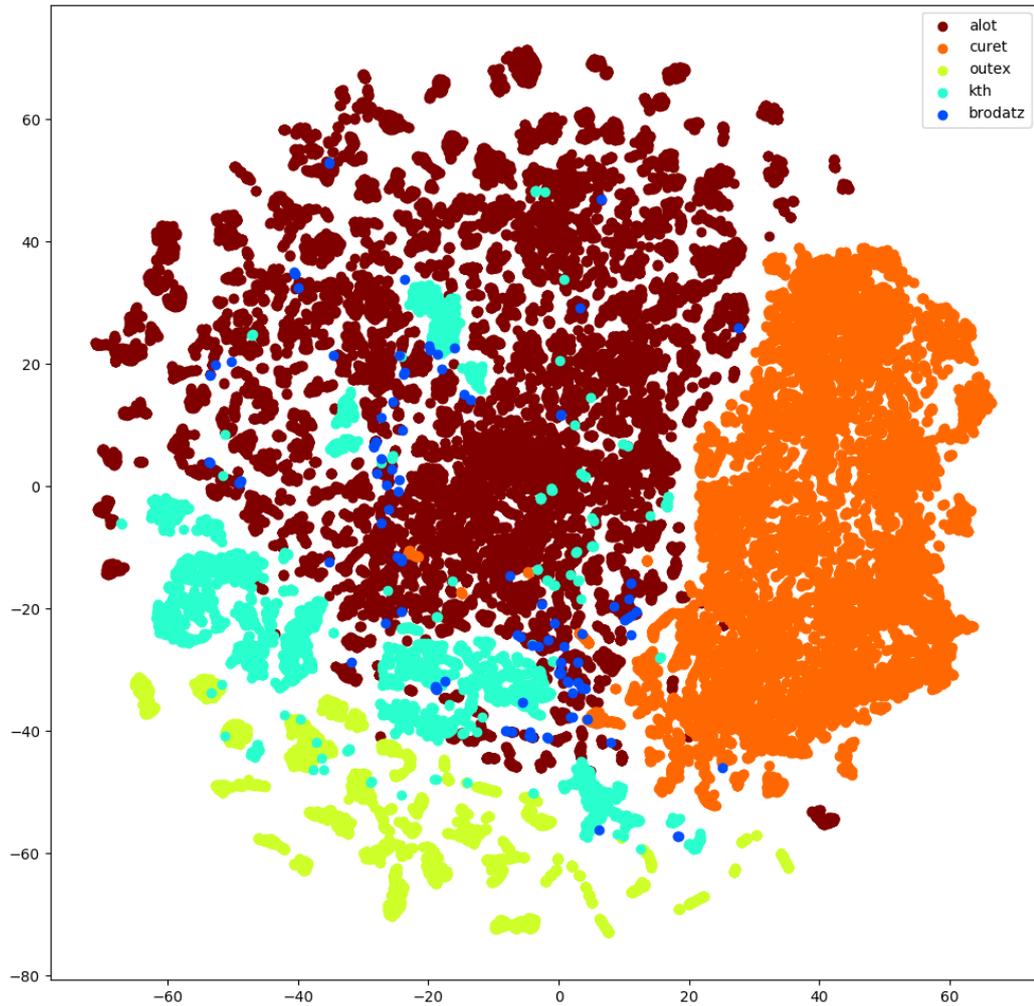
Figura 35 – Etapas do framework proposto.



Fonte: Elaborado pelo autor, 2019.

Contudo, o espaço de textura é gerado considerando todas as dimensões que existem nos vetores de características. Ficaria impraticável a visualização desse espaço sem ser utilizada um método que facilite isso. O *framework* utiliza o método *t-Distributed Stochastic Neighbor Embedding*(t-SNE), que é uma técnica não linear para redução de dimensionalidade adequada para a visualização de conjuntos de dados de alta dimensão. Originalmente, o espaço gera características com 1000 dimensões e após a aplicação da técnica t-SNE ele é representado por duas dimensões. A Figura 36 mostra como a visualização do espaço após o processamento de uma rede *Triplet* e o t-SNE.

Figura 36 – Dimensões do espaço de texturas reduzidas pelo t-SNE.



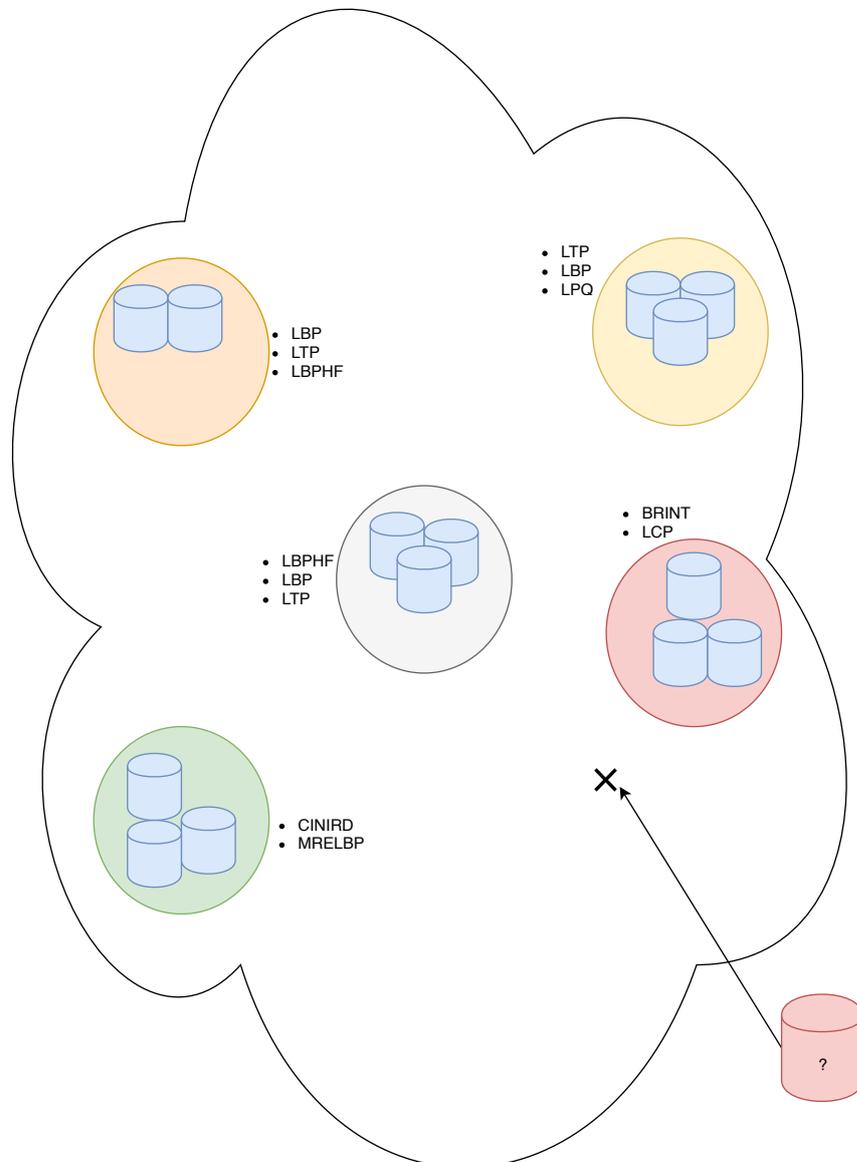
Fonte: Elaborado pelo autor, 2019.

O *framework* utiliza resultados de classificação já obtidos previamente para cada base: os experimentos de Liu et al. (2017) mostram uma grande fonte de descritores aplicados em bases de dados dentro do problema de classificação de texturas. O modo como os descritores foram abordados, bem como a metodologia dos experimentos mostram resultados que procuram refletir apenas a qualidade do descritor independentemente do classificador que é utilizado. Uma vez com o espaço gerado, uma base de dados que não foi utilizada para gerar o espaço é utilizada com o objetivo de calcularmos de qual região essa base se aproxima e, a partir das bases que estão nessa região, estimarmos quais técnicas de descrição e classificação são melhores para essa base desconhecida através do ranking de

classificação que recebe como entrada a distância de qual base do espaço textural é mais próxima da base de entrada.

O espaço textural é algo como a Figura 37. Todas as bases de treinamento ganham uma representação no espaço baseado na função que é descoberta pela métrica de similaridade. Quando uma base de dados desconhecida é avaliada usando o espaço de texturas, conseguimos descobrir qual a distância que ela tem de outra base de textura que já conhecemos os resultados, baseada em sua similaridade através de um *Ranking*.

Figura 37 – Espaço textural de texturas gerado pela métrica de similaridade.



3.2.1 Similaridade de Texturas

A geração do espaço de texturas precisa ser coerente. Esse espaço precisa, inevitavelmente, considerar a similaridade entre os dados de entrada e a partir disso distribuir no espaço as regiões que serão consideradas “regiões texturais”.

3.2.1.1 SSIM

Dadas duas imagens de entrada m e n , o *Structural Similarity* (SSIM) é usado para avaliar suas similaridades. O índice SSIM é uma métrica de referência completa, ou seja: a medição ou previsão da qualidade da imagem é baseada em uma imagem inicial descompactada, ou sem distorção como referência. O SSIM foi projetado para melhorar os métodos tradicionais, como a *peak signal-to-noise ratio* (PSNR) e o erro médio quadrático (EMQ). A fórmula do SSIM é como mostra a equação 3.1:

$$SSIM = \frac{(2u_m u_n + c_1)(2\sigma_{mn} + c_2)}{(u_m^2 + u_n^2 + c_1)(\sigma_m^2 + \sigma_n^2 + c_2)} \quad (3.1)$$

onde u_m e u_n são as médias de m e n ; σ_m^2 e σ_n^2 são as variâncias de m e n ; σ_{mn} é a covariância de m e n e por fim, c_1 e c_2 são variáveis que estabilizam a divisão com o operador fraco da fórmula. Os SSIMs podem ser aplicados no domínio espacial ou de transformação. Quando implementada no domínio da imagem, a métrica SSIM é invariante para iluminar e contrastar as alterações, mas é sensível à conversão, dimensionamento e rotação de imagens. Um experimento de testes foi executado com o intuito de provar a possível viabilidade de uso do SSIM como métrica de similaridade no *framework*.

Esse experimento de testes aborda a aplicação do SSIM em uma base de dados para verificar a resposta que ele tem como métrica de similaridade. Foram abordados os experimentos verificando imagens de mesma classe e todas as imagens da base de dados em uma comparação n para n .

Para esse teste foi utilizado apenas uma classe da base da UMD, que é uma das bases que serão utilizadas no experimento total. A base da UMD possui um total de 25 classes com 40 imagens em escala de cinza por classe com dimensões 1280×960 .

Os resultados da Tabela 3 mostram os valores máximos, médios e mínimos para cada uma das classes da base de dados. Como podemos ver, é impraticável o uso do SSIM para esse problema. Mesmo com exemplos que pertencem a mesma classe de texturas, os resultados mostraram que essa métrica não é própria para identificar similaridades entre as minúcias das texturas.

Tabela 3 – Experimento utilizando SSIM como métrica de similaridade de textura dentro de uma das 25 classes da base UMD.

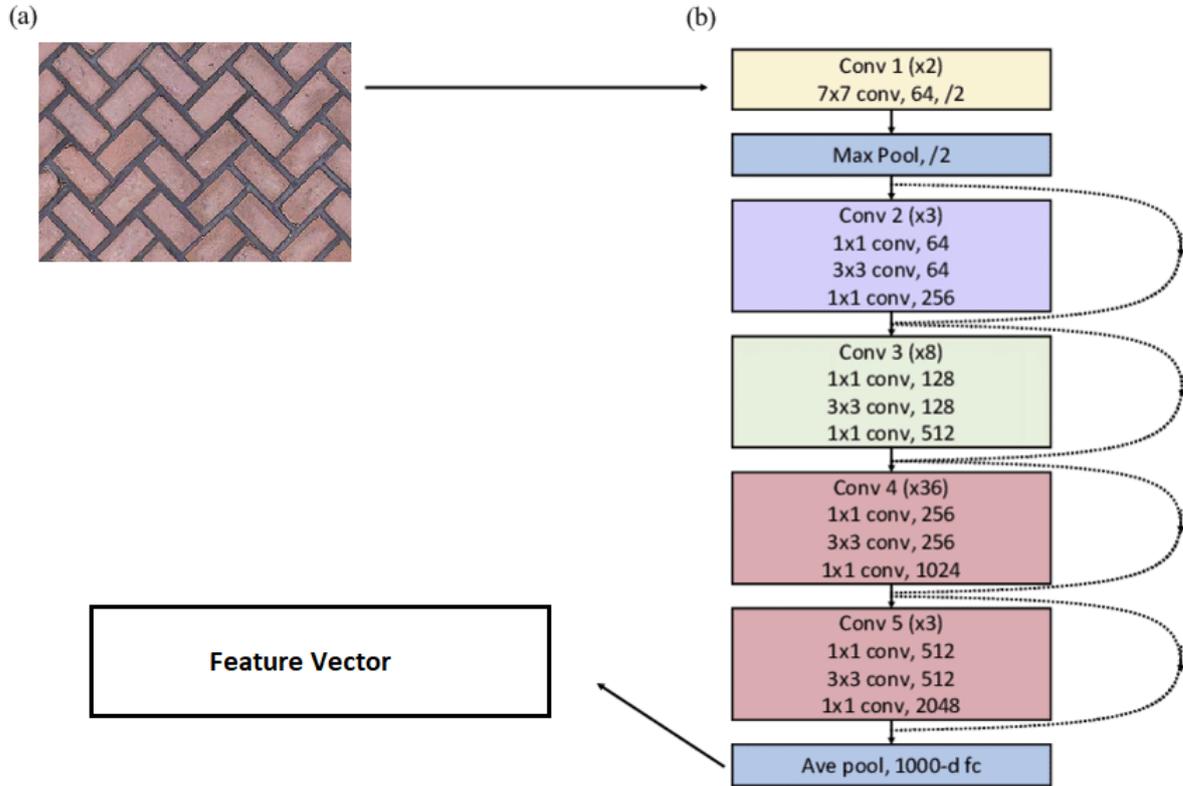
Classe	Média	Max	Min
1	0,1077	0,2069	0,0759
2	0,2979	0,5992	0,2071
3	0,0730	0,1665	0,0318
4	0,2064	0,3318	0,0999
5	0,5692	0,6852	0,3874
6	0,1555	0,2696	0,0826
7	0,2426	0,3419	0,1482
8	0,2022	0,3091	0,1406
9	0,4067	0,5092	0,2864
10	0,1324	0,2615	0,0752
11	0,1096	0,1587	0,0800
12	0,2130	0,4809	0,0580
13	0,1223	0,2241	0,0638
14	0,1710	0,2809	0,1109
15	0,0251	0,0492	0,0153
16	0,0335	0,0923	0,0118
17	0,2491	0,4361	0,1452
18	0,1740	0,4962	0,0544
19	0,1166	0,1919	0,0799
20	0,0845	0,2211	0,0278
21	0,0711	0,1301	0,0411
22	0,4106	0,5270	0,2899
23	0,0507	0,0854	0,0327
24	0,1161	0,3034	0,0726
25	0,0401	0,0727	0,0252

Fonte: Elaborado pelo autor, 2019.

3.2.1.2 Triplet loss function e Redes Convolucionais

A primeira aplicação prática da *Triplet loss function* foi feita para o problema de reconhecimento facial (SCHROFF; KALENICHENKO; PHILBIN, 2015). A rede *Triplet* foi feita com base na rede siamesa e é composta por 3 instâncias da mesma rede *feedforward*, onde seus parâmetros são compartilhados. Essa rede então recebe como entradas 3 imagens para treino. Essas imagens são então incorporadas em um vetor de características gerado por uma rede convolucional (CNN) com pesos compartilhados entre si. Essa rede CNN pode utilizar qualquer modelo CNN disponível. A Figura 38 mostra uma ResNet sendo utilizada como extrator de características.

Figura 38 – Resnet sendo utilizada como extrator de características - a) Imagens de entrada da rede, b) Camadas da ResNet152 utilizadas para extrair o vetor de características.



Fonte: (SCHROFF; KALENICHENKO; PHILBIN, 2015).

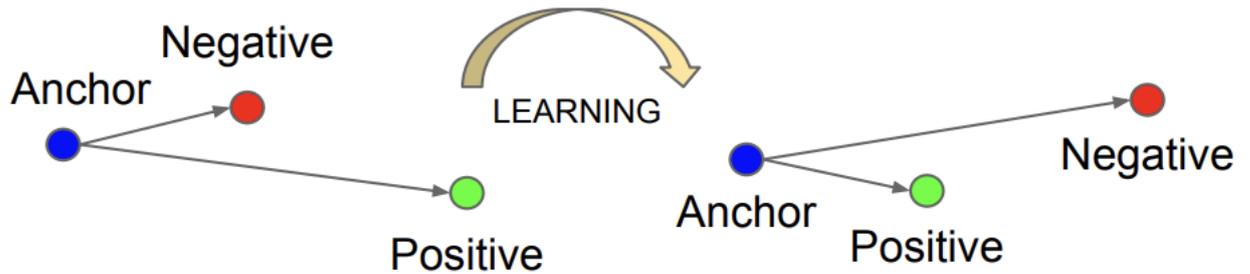
É criado um critério que mede a perda do trio do *batch* de entrada e uma margem com um valor maior que 0. Isso é usado para medir uma similaridade relativa entre as amostras do *batch*. Com isso, são produzidos valores intermediários. Esses valores são as distâncias L2 entre a representação embutida de duas de suas entradas a representação do terceiro. Essa distância é o cálculo que comumente é utilizada como mostrada na Equação 2.23 mostrada no Capítulo 2. A equação 3.2 denota às três entradas como sendo x , x^+ e x^- . A Figura 40 mostra a arquitetura da *triplet*.

$$TripletNet(x, x^+, x^-) = \begin{bmatrix} ||Net(x) - Net(x^-)||_2 \\ ||Net(x) - Net(x^+)||_2 \end{bmatrix} \in \mathbb{R}_+^2 \quad (3.2)$$

O treinamento da rede *Triplet* é realizado tendo como entradas x , x^+ e x^- onde x e x^+ são da mesma classe e x^- é um exemplo negativo da classe. A arquitetura da rede transforma a tarefa em uma estrutura igual a um problema de classificação de duas classes: classificar corretamente a qual classe x^+ e x^- pertencem baseando-se em x . Essas imagens são incorporadas em um espaço euclidiano com d-dimensões, ou seja, um vetor de características. Quando aplicada a uma descoberta de métrica, o rótulo define qual

das entradas está mais próximo de x onde o conceito de próximo define que os exemplos são de mesma classe, ou seja, compartilham o mesmo rótulo. Com isso, função *triplet loss* penaliza a classe negativa de forma que a distância entre os exemplos correspondentes seja reduzida e a distância entre as não correspondentes aumentada. A Figura 39 mostra o efeito da função no elemento x de entrada, normalmente chamado de elemento “Âncora”.

Figura 39 – Efeito da função *triplet loss* nos elementos de treinamento.



Fonte: (SCHROFF; KALENICHENKO; PHILBIN, 2015).

O vetor de características é representado por $f(x) \in \mathbb{R}^d$, que representa a imagem x em um espaço euclidiano com d -dimensões. Com isso, temos

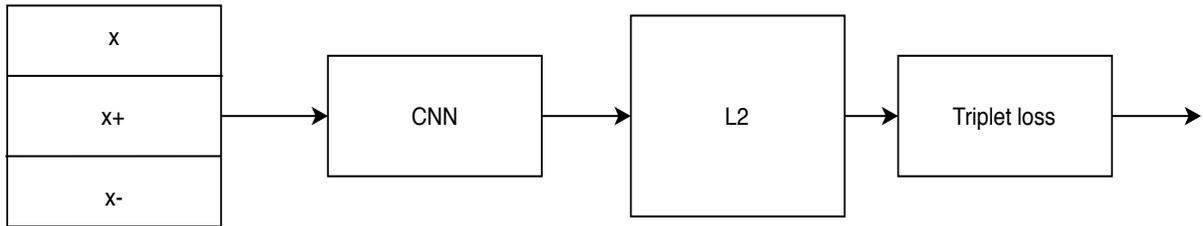
$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \quad (3.3)$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \tau \quad (3.4)$$

onde α é uma margem que é aplicada entre exemplos positivos e negativos. τ é o conjunto de todos os *triplets* possíveis no conjunto de treinamento com cardinalidade N .

A resultante disso é um vetor de características que será suficiente para a separação, uma vez que exemplos similares vão possuir distâncias pequenas podendo ser agrupados e exemplos diferentes vão possuir distâncias maiores que permitem a discriminação entre eles. Essa abordagem foi usada como base para o trabalho de Schroff, Kalenichenko e Philbin (2015) que alcançou resultados do estado da arte em bases de reconhecimento facial.

Figura 40 – Arquitetura da rede *triplet network*, onde x é uma imagem da classe ancora, $x+$ um exemplo positivo dessa classe e $x-$ um exemplo negativo.



Fonte: Elaborado pelo autor, 2019.

Com essa definição, vemos uma possibilidade da criação do espaço de texturas com a utilização da *triplet network* como modelo de similaridade, já que a mesma pode servir como uma metodologia que pode ser usada para separar as bases de dados diferentes no espaço e agrupar as similares.

4 EXPERIMENTOS E RESULTADOS

4.1 INTRODUÇÃO

Os testes foram elaborados visando avaliar a hipótese de que, uma vez que as bases sejam próximas no espaço elas vão possuir descritores de características semelhantes e é possível prever quais descritores funcionam melhor para ela e quais descritores tem um desempenho pior. Para isso, o experimento segue o fluxo do *framework* proposto na Seção 3.2. As bases consideradas para o experimento foram descritas na Seção 2.5. O trabalho de Liu et al. (2017) disponibiliza resultados produzidos pelos descritores de características para essas bases de dados. Esses resultados são utilizados para o *ranking* de descritores de cada base que servirá para mostrar os possíveis descritores candidatos para uma base desconhecida como é mostrado na Figura 35.

Neste capítulo também são mostradas três bases de testes que não são consideradas na criação do espaço sendo aplicadas ao espaço de texturas e a relação dessas bases com os melhores descritores do espaço ao qual elas pertencem.

4.2 CONFIGURAÇÃO DAS BASES DE DADOS E DO EXPERIMENTO

Os experimentos foram conduzidos utilizando as bases de dados mostradas na Tabela 2, incluindo bases de texturas com uma grande quantidade de imagens e classes diferentes de texturas. A avaliação tem, como mostrado na Seção 3.2, três elementos chave: a geração do espaço de texturas, a inclusão de uma base nova para ser feita a classificação, e a avaliação dos resultados através de um *ranking*. Antes da geração do espaço é feito um processamento da imagem de entrada com o objetivo de padronizá-las, que como mostrado Tabela 2, diferentes bases possuem diferentes categorias de dados com diferentes tamanhos e configurações. O pré-processamento tornam os exemplos em imagens 224×224 de tamanho.

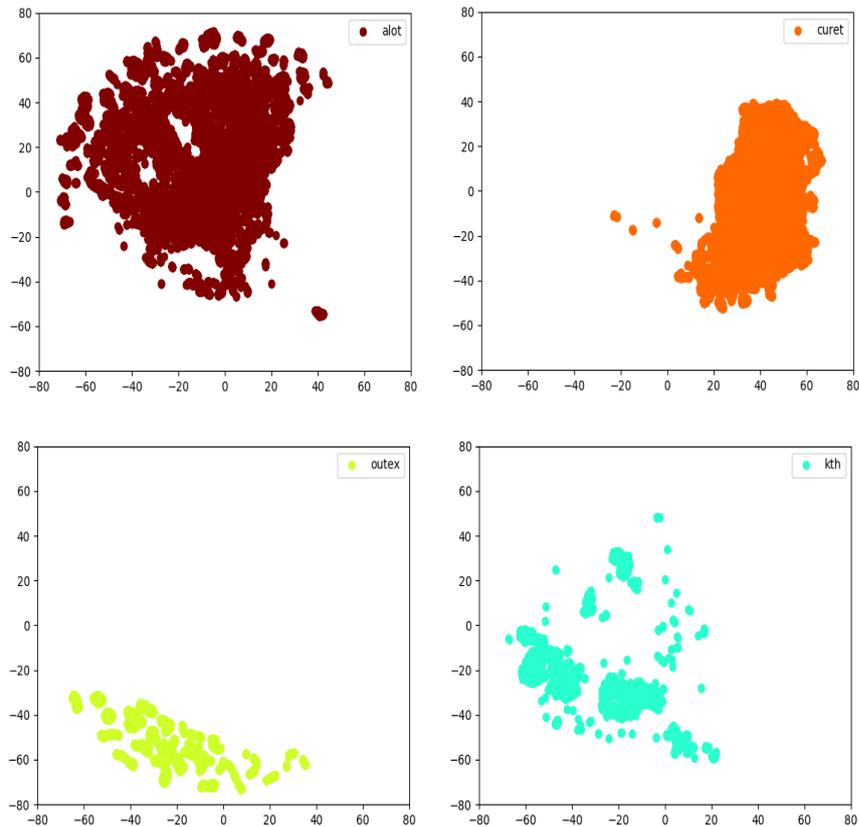
4.2.1 Espaço de texturas

Para gerar o espaço de texturas foram consideradas 4 bases de dados considerando a maior quantidade de imagem que elas possuem. Como mostrado na Tabela 2, as bases ALOT, CURET, KTH e Outex possuem uma maior quantidade de imagens e são consideradas para o primeiro protocolo de treinamento. Além de possuírem uma grande quantidade de imagens, também possuem um número diversificado de classes diferentes.

Como mostrado na Seção 3.2.1.2, a *Triplet Network* foi considerada para geração do espaço de texturas. Para tal, foi utilizado o modelo ResNet mostrado na Seção 2.3.6 como CNN's que servem como extratores de características. Foram consideradas 150 épocas de

treinamento. As Figuras 41 e 42 mostram as bases individualmente no espaço e o espaço formado com às quatro bases nessa configuração.

Figura 41 – Bases utilizadas no treinamento mostradas individualmente no espaço: ALOT, CURET, Outex e KTH.

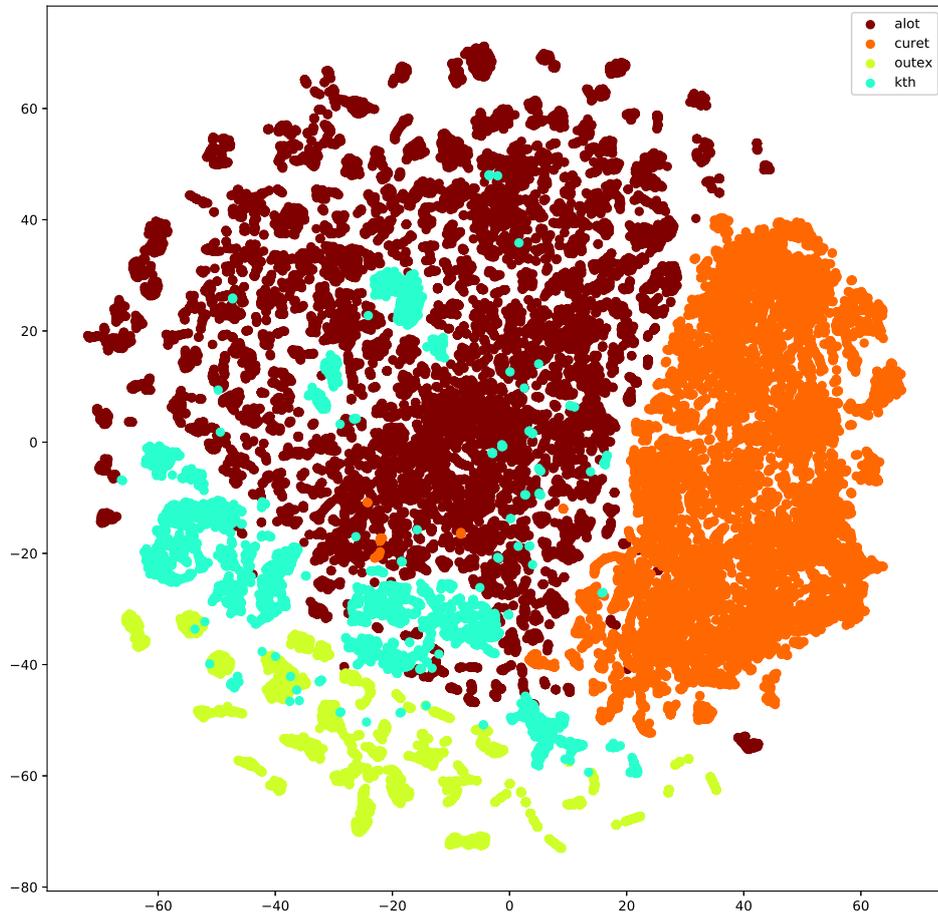


Fonte: Elaborado pelo autor, 2019.

4.2.2 Classificação de teste em uma base desconhecida

Uma vez que a rede tenha gerado o espaço de texturas através do treinamento citado na Seção anterior, utilizamos o mesmo para fazer a classificação de uma base de dados que não tenha sido utilizada no treino. Para esses testes, foram consideradas três bases de dados: UMD, UIUC e a Brodatz. A classificação é feita baseada nos resultados de distância de um classificador k-NN, considerando como configurações a distância euclidiana citada na Seção 2.4, número de k sendo 7 e a classificação uniforme. Foram executadas configurações para o número de k sendo 3, 5 e 15, contudo não tiveram diferença nos dados de saída. Para cada exemplo da base desconhecida será calculada a distância com um elemento do espaço. A base desconhecida de testes assume então uma região no espaço onde podemos calcular de quem ela está mais próxima baseada no histograma de todas as distâncias.

Figura 42 – Espaço formado pelas bases ALOT, CURET, Outex e KTH.



Fonte: Elaborado pelo autor, 2019.

Para cada uma das bases foram gerados os histogramas dos resultados bem como sua distribuição no espaço.

4.3 AVALIAÇÕES

Os resultados dos testes foram avaliados individualmente por base testada na camada de *ranking*. Para cada uma das bases de dados foram considerados os melhores 16 e 16 piores resultados dos descritores locais do trabalho de Liu et al. (2017). Os resultados para cada uma das bases são mostrados nas Tabelas 4, 5, 6 e 7. Esses resultados consideram a medida de Acurácia. A medida de Acurácia é dada pela Equação 4.1, que fornece a proporção de exemplos classificados corretamente.

$$Acurácia = \frac{v_p + v_n}{v_p + f_p + v_n + f_n} \quad (4.1)$$

Para cada uma das bases de testes, são comparadas a quantidade de melhores descritores que ela possui com sua região de similaridade.

Tabela 4 – Os melhores descritores de texturas locais baseados em sua acurácia de classificação para ALOT, Brodatz, CURET e Kth. O campo "*"significa que para a base ALOT existem apenas 15 melhores.

ALOT		Brodatz		CURET		kth	
BRINT	96,13	BSIF	94,48	BRINT	97,02	BRINT	66,67
CINIRD	97,21	CINIRD	93,24	BSIF	96,83	BSIF	63,74
CLBC	96,46	CLBP	92,34	CINIRD	96,60	CINIRD	64,84
CLBP	96,74	CLBPHF	91,95	CLBC	96,86	CLBP	64,18
CLBPHF	96,30	disCLBP	93,18	CLBP	97,33	CLBPHF	68,10
disCLBP	95,01	ILBP	91,66	CLBPHF	97,05	disCLBP	63,83
EXLBP	95,20	LBPHF	92,56	disCLBP	96,98	LBP	62,69
ILBP	95,71	LFD	94,92	EXLBP	96,85	LBPHF	67,51
LBPHF	94,91	LoQP	91,02	LBPHF	97,51	LBP	62,69
LTP	94,60	LTP	92,41	LBP	97,03	LEP	63,13
MDLBP	95,81	MDLBP	93,40	LFD	97,39	LoQP	63,74
MRELBP	97,28	MRELBP	90,86	MDLBP	96,92	LTP	63,45
MSJLBP	95,65	MSJLBP	92,94	MRELBP	97,10	MDLBP	66,52
RILPQ	94,85	PRICoLBP	92,94	MSJLBP	97,20	MRELBP	68,98
SSLBP	96,68	RILPQ	91,37	RLBP	97,20	MSJLBP	65,51
*	*	RLBP	91,09	SSLBP	98,79	SSLBP	65,57

Fonte: Elaborado pelo autor, 2019.

Tabela 5 – Os melhores descritores de texturas locais baseados em sua acurácia de classificação para Outex, UIUC e UMD.

Outex		UIUC		UMD	
CINIRD	99,66	BRINT	93,30	BSIF	96,15
CLBC	99,40	CINIRD	94,61	disCLBP	97,49
CLBP	99,45	CLBC	94,37	EXLBP	98,93
CLBPHF	99,69	CLBP	95,75	FLBP	97,26
disCLBP	99,95	CLBPHF	92,55	ILBP	96,78
DLBP	99,45	disCLBP	94,24	LBP/VAR 2	98,66
EXLBP	99,64	EXLBP	88,29	LBPV	96,66
ILBP	99,66	ILBP	92,84	LFD	96,54
LBPHF	99,69	LBDP	90,98	LTP	98,4
LBP	99,69	LBPHF	89,58	MBP	97,36
LoQP	99,64	LBP	88,36	MDLBP	97,24
LTP	99,92	LoQP	93,20	MRELBP	98,41
MRELBP	99,82	LTP	93,27	MSJLBP	96,53
RILPQ	99,58	MRELBP	94,73	NTLBP 29	97,44
RLBP	99,66	RILPQ	91,17	RLBP	97,53
SSLBP	99,82	SSLBP	95,40	SSLBP	98,62

Fonte: Elaborado pelo autor, 2019.

Tabela 6 – Os piores descritores de texturas locais baseados em sua acurácia de classificação para as Bases ALOT, Brodatz, CURET e Kth. O campo "*"significa que para a base ALOT existem apenas 14 piores.

ALOT		Brodatz		CURET		kth	
BSIF	93,09	BRINT	90,83	DLBP	94,38	CLBC	58,27
LBDP	92,82	CLBC	90,24	FLBP	95,38	DLBP	61,72
LBPri2	93,29	DLBP	88,73	ILBP	94,66	EXLBP	62,39
LBP	94,15	EXLBP	90,19	LBDP	94,23	FLBP	62,50
LBP/VAR 2	93,62	FLBP	89,02	LBPri2	95,38	ILBP	61,93
LBPV	91,87	LBDP	89,74	LBP/VAR 2	96,04	LBPri2	62,48
LEP	89,67	LBPri2	89,93	LBPV	95,69	LBP/VAR 2	58,52
LFD	89,11	LBP	90,70	LBPV	95,85	LBPV	59,43
LoQP	94,31	LBP/VAR 2	88,71	LEP	88,31	LBPV	59,03
MBP	88,23	LBPV	89,07	LoQP	95,73	LFD	56,65
NRLBP	87,86	LBPV	87,63	LTP	96,33	MBP	61,49
NTLBP 29	94,47	LEP	82,64	MBP	92,09	NRLBP	58,61
PRICoLBP	94,38	MBP	87,25	NRLBP	94,00	NTLBP 29	61,30
RLBP	94,23	NRLBP	87,42	NTLBP 29	96,11	PRICoLBP	61,17
*	*	NTLBP 29	89,31	PRICoLBP	96,25	RILPQ	58,75
*	*	SSLBP	89,94	RILPQ	92,15	RLBP	61,20

Fonte: Elaborado pelo autor, 2019.

Tabela 7 – Os piores descritores de texturas locais baseados em sua acurácia de classificação para as Bases Outex , UIUC e UMD.

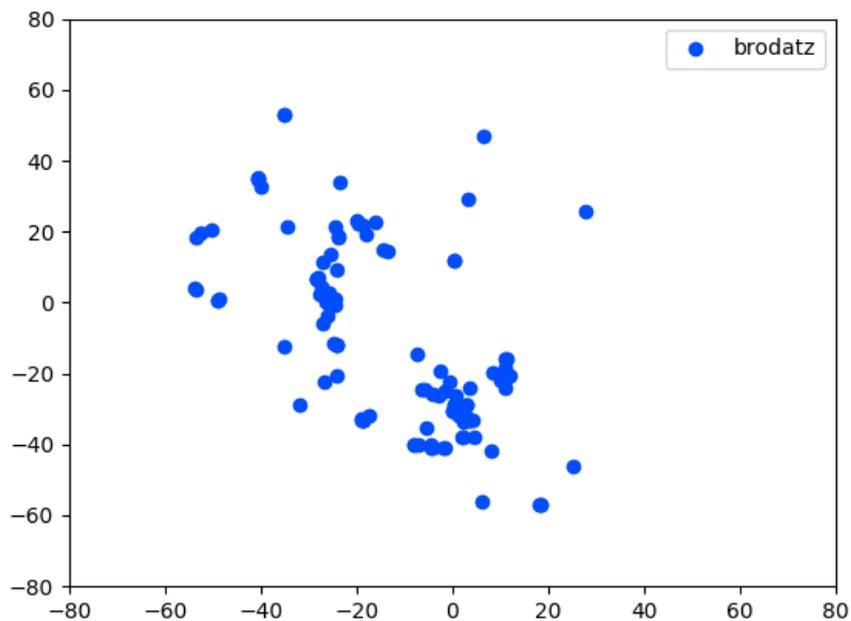
Outex		UIUC		UMD	
BRINT	99,35	BSIF	73,39	BRINT	95,69
BSIF	57,68	DLBP	83,71	CINIRD	92,89
FLBP	98,78	FLBP	84,03	CLBC	93,79
LBDP	98,78	LBPri2	79,03	CLBP	96,10
LBPri2	86,69	LBP/VAR 2	84,44	CLBPHF	95,88
LBP/VAR 2	98,52	LBPV	80,74	DLBP	93,32
LBPV	99,22	LBPV	81,98	LBDP	94,89
LBPV	99,27	LEP	81,80	LBPHF	93,80
LEP	81,90	LFD	87,28	LBPri2	92,41
LFD	99,38	MBP	80,89	LBP	93,58
MBP	95,29	MDLBP	88,05	LBPV	95,55
MDLBP	99,22	MSJLBP	83,00	LEP	94,99
MSJLBP	96,67	NRLBP	81,10	LoQP	92,99
NRLBP	98,07	NTLBP 29	88,13	NRLBP	95,50
NTLBP 29	99,32	PRICoLBP	80,38	PRICoLBP	95,72
PRICoLBP	94,48	RLBP	88,29	RILPQ	91,75

Fonte: Elaborado pelo autor, 2019.

4.3.1 Brodatz

Após a projeção da base Brodatz no espaço mostrado na Figura 43, a Brodatz ocupa um espaço dominado pela base ALOT, como mostra o gráfico da Figura 44. Os histogramas da classificação confirmam essa maior afinidade visual da Brodatz com a ALOT, como mostra a Figura 46, com uma quantidade de exemplos dominante quando consideramos as demais regiões.

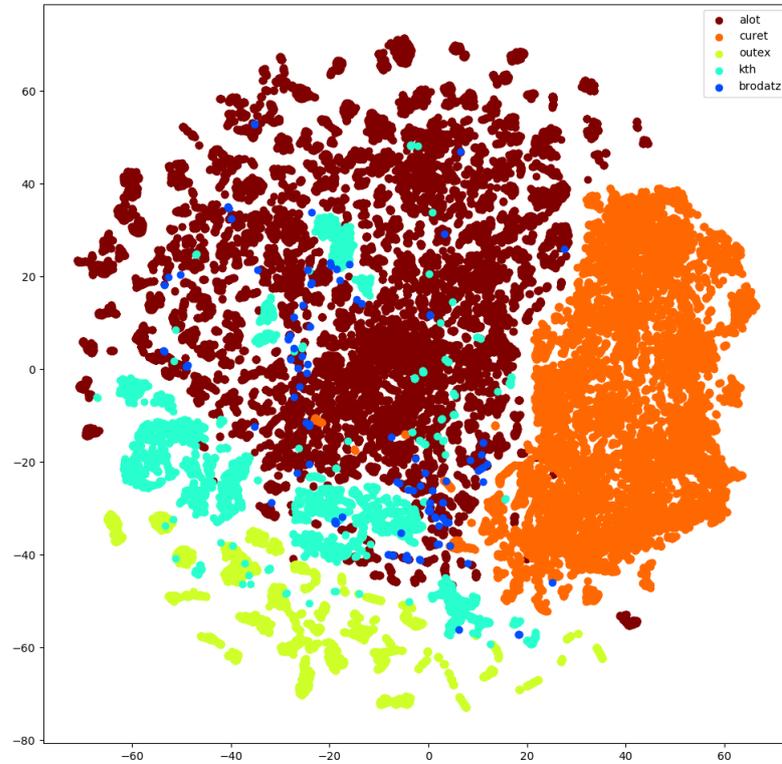
Figura 43 – Projeção individual da base de testes Brodatz no espaço de texturas



Fonte: Elaborado pelo autor, 2019.

As avaliações revelam que, a Brodatz e a ALOT compartilham 80% dos descritores de textura que são considerados os melhores quando são consideradas as taxas de classificação da tabela. As taxas de classificação também são semelhantes: quando consideramos, por exemplo, o descritor CINIRD em ambas as bases mantêm uma taxa acima de 90% como mostra a Tabela 8.

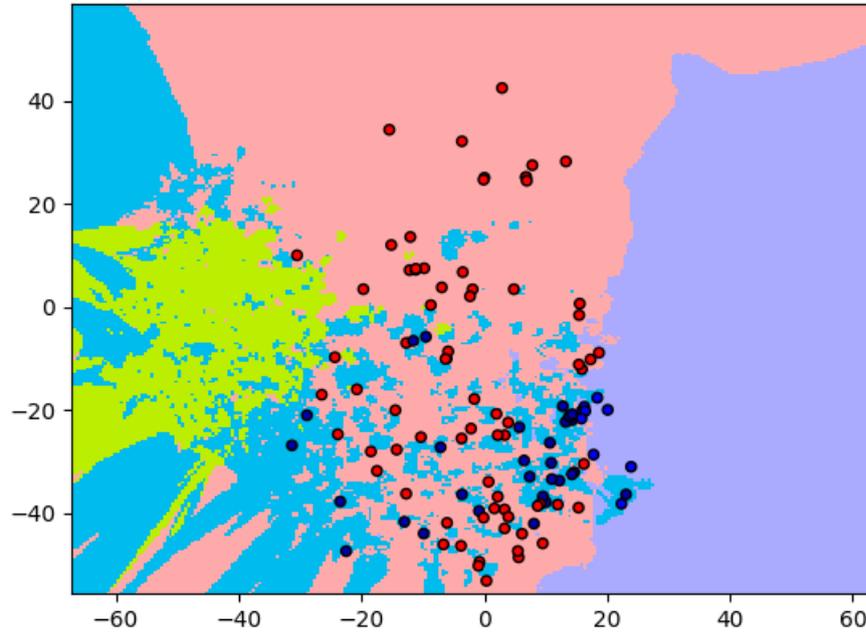
Figura 44 – Distribuição da base de testes Brodatz no espaço de texturas.



Fonte: Elaborado pelo autor, 2019.

Quando consideramos os piores descritores da tabela, as duas bases compartilham 65% dos descritores. O descritor de texturas LEP, presente na lista de piores descritores de ambas as bases, possui um desempenho inferior à maioria dos demais em ambas as bases. A lista de melhores descritores presentes em ambas as bases é mostrada na Tabela 8 com seus respectivos desempenhos.

Figura 45 – Visualização regional da distribuição da base de testes Brodatz no espaço de texturas com $k = 7$.



Fonte: Elaborado pelo autor, 2019.

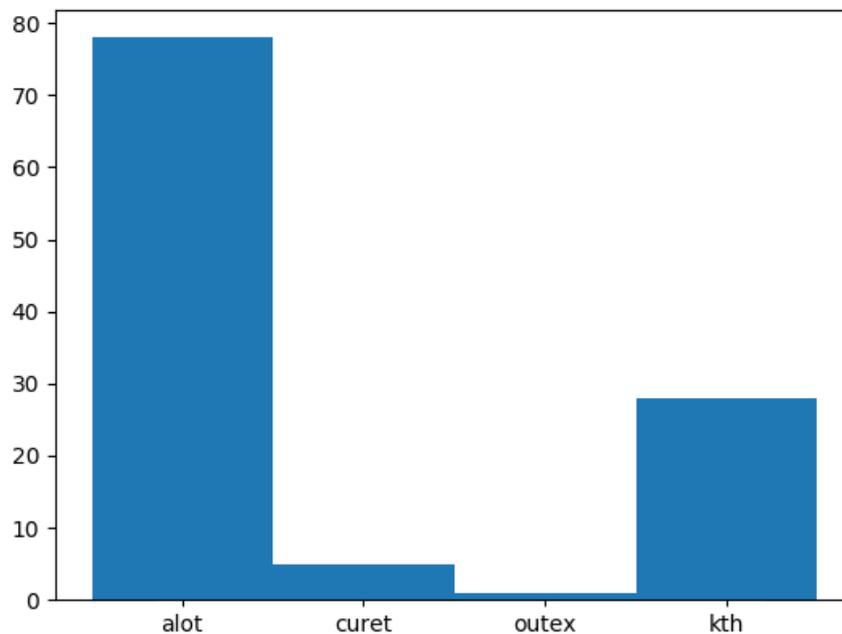
A visualização regional mostrada na Figura 45 confirma a maior afinidade da base de testes com a base selecionada pelo modelo. A ALOT é uma base dominante do espaço total e possui maioria dos exemplos. Essa distribuição é gerada considerando as saídas do modelo. As saídas, no entanto, são vetores com 1000 dimensões e para fazer alcançar uma representação confiável é aplicada uma técnica de redução de dimensões chamada TSNE (MAATEN; HINTON, 2008). Essa técnica é amplamente utilizada na literatura para visualizar dados que possuem muitas dimensões.

Tabela 8 – Melhores descritores presentes na relação ALOT e Brodatz.

Descritor	Resultado ALOT	Resultado Brodatz
CINIRD	97,21	93,24
CLBP	96,74	92,34
CLBPHF	96,30	91,95
disCLBP	95,01	93,18
ILBP	95,71	91,66
LBPHF	94,91	92,56
LTP	94,60	92,41
MDLBP	95,81	93,40
MRELBP	97,28	90,86
MSJLBP	95,65	92,94
RILPQ	94,85	91,37

Fonte: Elaborado pelo autor, 2019.

Figura 46 – Histograma de classificação do k-NN da base brodatz no espaço de texturas.



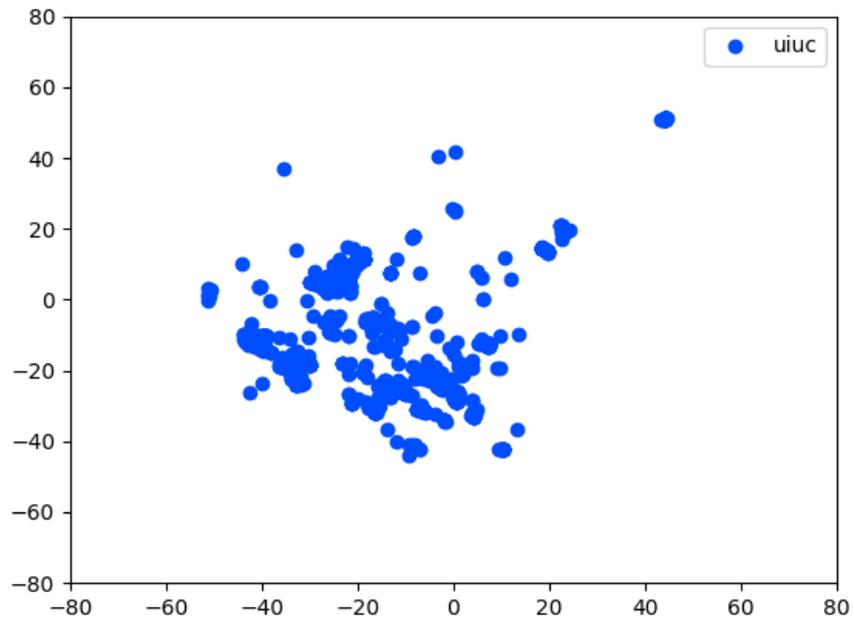
Fonte: Elaborado pelo autor, 2019.

4.3.2 UIUC

Após a projeção da base UIUC no espaço, mostrada na Figura 47, a base de dados UIUC também ocupa um espaço dominado pela base ALOT, como mostra o gráfico da Figura

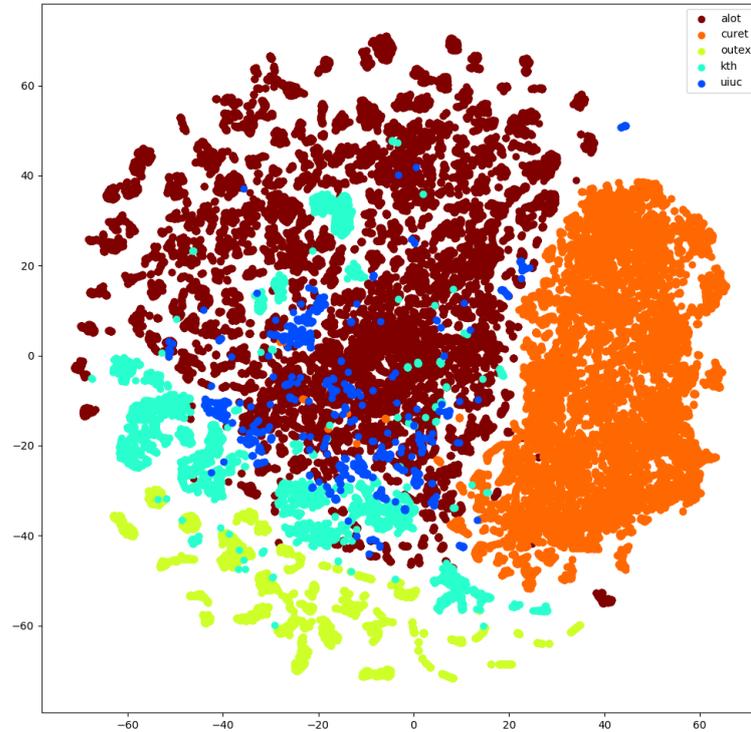
48. Os histogramas da classificação confirmam essa maior semelhança da base UIUC com a base ALOT, como mostra a Figura 50.

Figura 47 – Projeção individual da base de testes UIUC no espaço de texturas.



Fonte: Elaborado pelo autor, 2019.

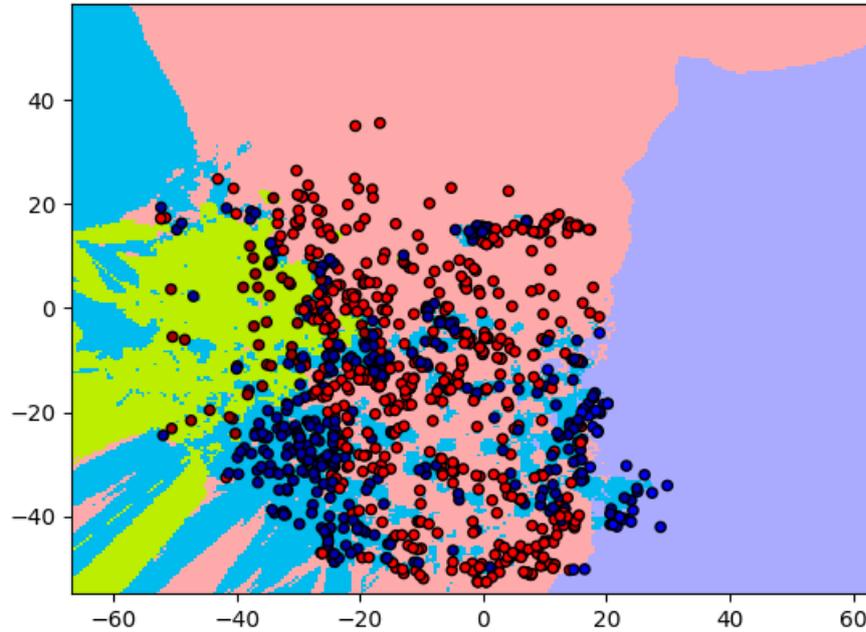
Figura 48 – Distribuição da base de testes UIUC no espaço de texturas.



Fonte: Elaborado pelo autor, 2019.

As avaliações revelam que a UIUC e a ALOT compartilham 87% dos descritores de textura que são considerados os melhores quando são consideradas as taxas de classificação da Tabela 4. Assim como a Brodatz taxas de classificação também são semelhantes. Quando consideramos, por exemplo, o descritor CLBP, em ambas as bases mantêm uma taxa acima de 90% como mostra a Tabela 9.

Figura 49 – Visualização regional da distribuição da base de testes UIUC no espaço de texturas com $k = 7$.



Fonte: Elaborado pelo autor, 2019.

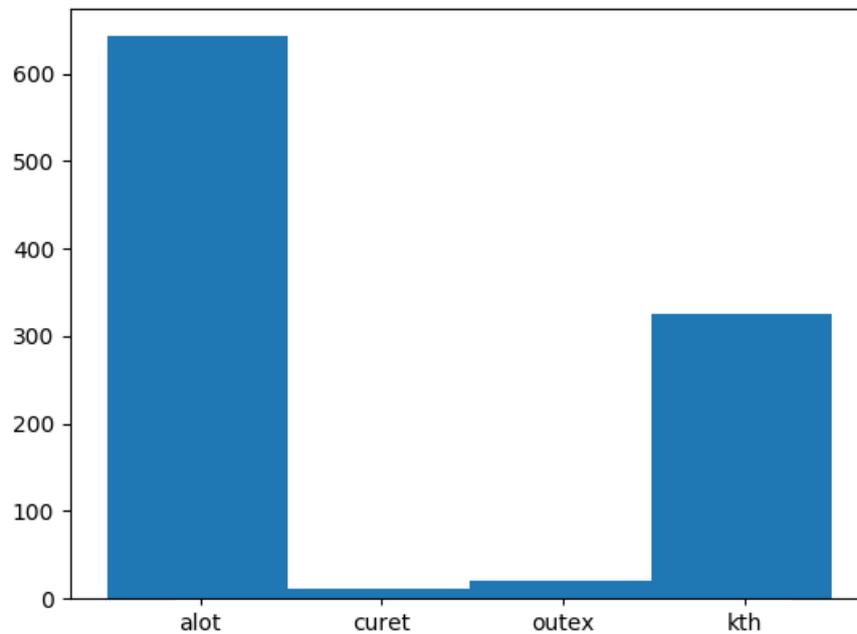
Quando consideramos os piores descritores da tabela, as duas bases compartilham 85% dos descritores. O descritor de texturas MBP está presente na lista de piores descritores de ambas as bases, possui um desempenho inferior à maioria dos demais em ambas as bases. A lista de melhores descritores presentes em ambas as bases é mostrada na Tabela 8 com seus respectivos desempenhos.

Tabela 9 – Melhores descritores presentes na relação ALOT e UIUC.

Descritor	Resultado ALOT	Resultado UIUC
BRINT	96,13	93,30
CINIRD	97,21	94,61
CLBC	96,46	94,37
CLBP	96,74	95,75
CLBPHF	96,30	92,55
disCLBP	95,01	94,24
EXLBP	95,20	88,29
ILBP	95,71	92,84
LBPHF	94,91	89,58
LTP	94,60	93,27
MRELBP	97,28	94,73
RILPQ	94,85	91,17
SSLBP	96,68	95,40

Fonte: Elaborado pelo autor, 2019.

Figura 50 – Histograma de classificação do k-NN da base UIUC no espaço de texturas.

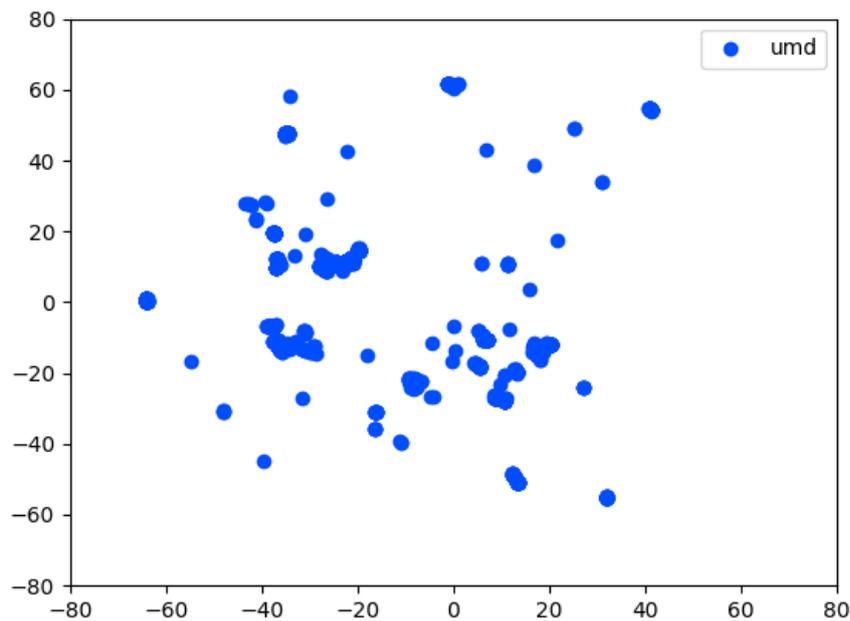


Fonte: Elaborado pelo autor, 2019.

4.3.3 UMD

Após a projeção da base UIUC no espaço, mostrada na Figura 51, a base de dados UIUC também ocupa um espaço dominado pela base ALOT, como mostra o gráfico da Figura 52. Os histogramas da classificação confirmam essa maior semelhança da base UIUC com a base ALOT, como mostra a Figura 54.

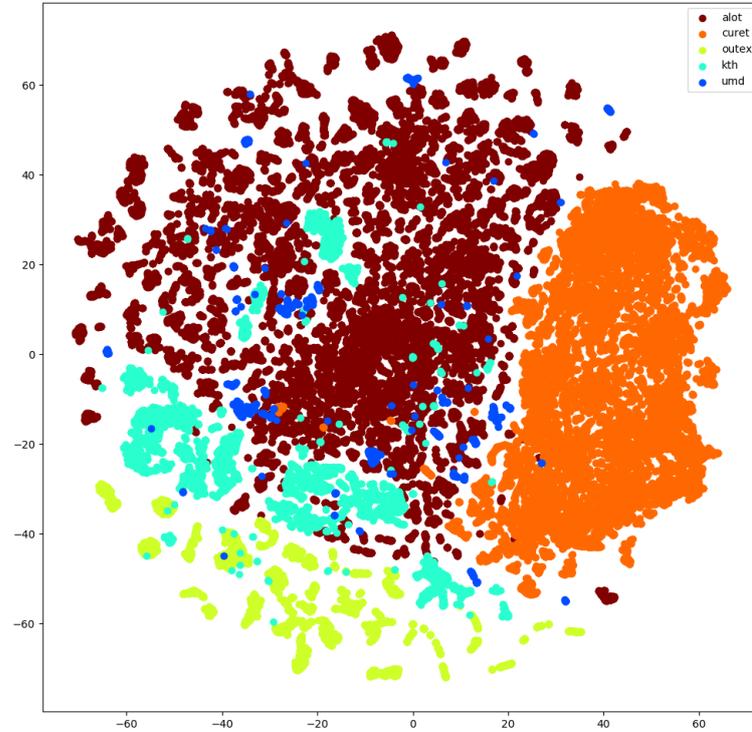
Figura 51 – Projeção individual da base de testes UMD no espaço de texturas.



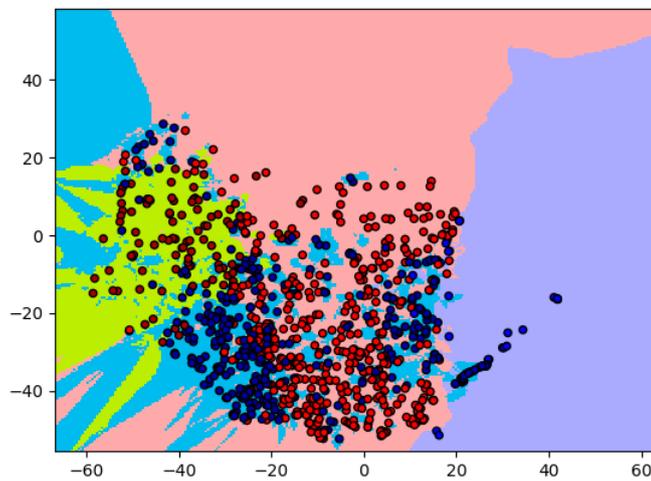
Fonte: Elaborado pelo autor, 2019.

As avaliações revelam que a UMD e a ALOT compartilham 55% dos descritores de textura que são considerados os melhores quando são consideradas as taxas de classificação da Tabela 4. Assim como a Brodatz taxas de classificação também são semelhantes. Quando consideramos, por exemplo, o descritor EXLBP, em ambas as bases mantêm uma taxa acima de 90% como mostra a Tabela 10.

Figura 52 – Distribuição da base de testes UMD no espaço de texturas.



Fonte: Elaborado pelo autor, 2019.

Figura 53 – Visualização regional da distribuição da base de testes UMD no espaço de texturas com $k = 7$.

Fonte: Elaborado pelo autor, 2019.

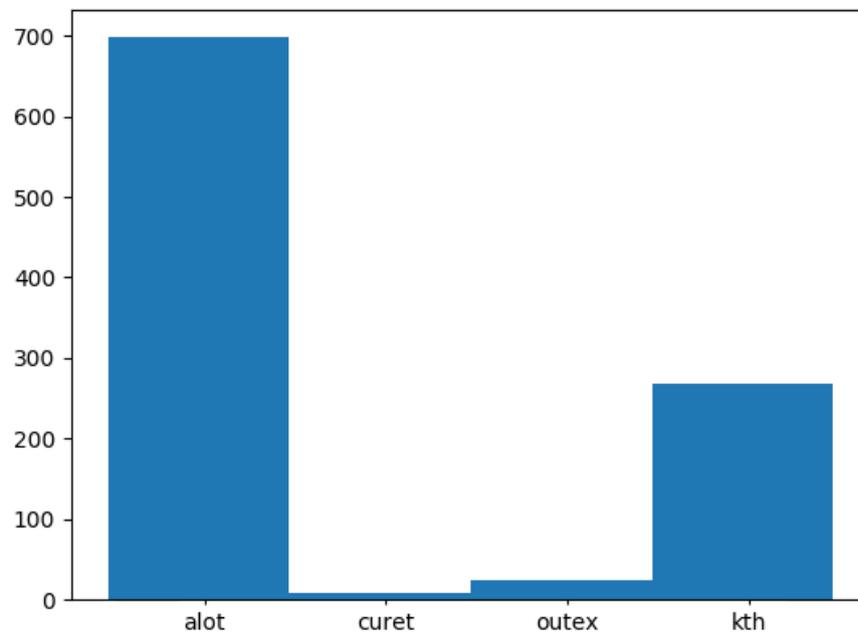
Quando consideramos os piores descritores da tabela, as duas bases compartilham 55% dos descritores. A lista de melhores descritores presentes em ambas as bases é mostrada na Tabela 10 com seus respectivos desempenhos.

Tabela 10 – Melhores descritores presentes na relação ALOT e UMD.

Descritor	Resultado ALOT	Resultado UMD
disCLBP	95,01	97,49
EXLBP	95,20	98,93
ILBP	95,71	96,78
LTP	94,60	98,4
MDLBP	95,81	97,24
MRELBP	97,28	98,41
MSJLBP	95,65	96,53
SSLBP	96,68	98,62

Fonte: Elaborado pelo autor, 2019.

Figura 54 – Histograma de classificação do k-NN da base UMD no espaço de texturas.



Fonte: Elaborado pelo autor, 2019.

5 CONCLUSÃO

A evolução da tecnologia foi algo extremamente benéfico para a tarefa de classificar texturas. O trabalho mostra que essa tarefa, que por muitas vezes é bastante exaustiva devido à quantidade de técnicas e dados, pode ser simplificada utilizando as informações que o modelo propõe. A geração do espaço de características de texturas e a capacidade de prever quais técnicas funcionariam e quais falhariam em uma base desconhecida através da sua similaridade com as regiões no espaço são as ferramentas propostas para alcançar tal feito. Primeiramente, o espaço de texturas que é gerado simplifica não só a visualização, mas também a organização dos dados de texturas. Além disso, somente a geração desse espaço de texturas mostra que, por mais confuso que seja a organização das classes de texturas nas bases da literatura, é possível separar tais bases em regiões texturais. Por fim, o modelo também mostra que podemos utilizar esse espaço de texturas para prever quais descritores de características falham e quais tem sucesso em uma determinada base desconhecida. Foi mostrado que, a maior parte dos descritores que tiveram um desempenho em uma determinada região puderam repetir esse comportamento nas bases que foram consideradas similares a essa região. A geração do espaço de texturas e a capacidade de prever quais técnicas funcionariam e quais falhariam em uma base desconhecida através da sua similaridade com as regiões no espaço dão suporte a uma otimização no tempo e no esforço nesse campo. Não apenas isso, mas também mostram uma possibilidade de separação e organização dos dados de texturas existentes, como em uma tabela periódica dos elementos da área química.

5.1 TRABALHOS FUTUROS

O estudo foi executado em uma quantidade relevante de bases de texturas. Contudo, a quantidade de bases diferentes em trabalhos futuros pode ser aumentada para termos uma região textural mais populada. A análise desse trabalho foi considerada para o estudo das bases de dados de texturas clássicas. Aplicações práticas podem surgir a partir dos resultados que o modelo foi capaz de gerar, por exemplo, aplicar a predição do espaço direcionado a um problema real dentro de áreas como medicina, robótica, etc. O modelo também pode ser aplicado, salvo ajustes, para ao invés de considerarmos similaridade entre bases, considerarmos similaridade entre as classes independentemente da base de dados que elas estejam. Para esse trabalho, no entanto, a reorganização manual de uma quantidade alta de dados deve ser feita bem como um modelo de classificação padrão para os métodos de descrição para cada uma dessas classes.

REFERÊNCIAS

- AHONEN, T.; MATAS, J.; HE, C.; PIETIKÄINEN, M. Rotation invariant image description with local binary pattern histogram fourier features. Springer, p. 61–70, 2009.
- ALEX, K.; ILYA, S.; HG, E. Imagenet classification with deep convolutional neural networks. *Proceedings of NIPS, IEEE, Neural Information Processing System Foundation*, p. 1097–1105, 01 2012.
- ARMI, L.; ERSHAD, S. F. Texture image analysis and texture classification methods - a review. *International Online Journal of Image Processing and Pattern Recognition*, 04 2019.
- BRODATZ, P. Textures: a photographic album for artists and designers, by phil brodatz. Dover publications, 1966.
- BURGHOUTS, G. J.; GEUSEBROEK, J.-M. Material-specific adaptation of color invariant features. *Pattern Recognition Letters*, v. 30, p. 306–313, 2009.
- CAPUTO, B.; HAYMAN, E.; MALLIKARJUNA, P. Class-specific material categorisation. In: . [S.l.: s.n.], 2005. v. 2, p. 1597 – 1604 Vol. 2. ISBN 0-7695-2334-X.
- CASTELLANO, G.; BONILHA, L.; LI, L.; CENDES, F. Texture analysis of medical images. *Clinical Radiology*, v. 59, p. 1061–1069, 2004.
- CHAKRABORTI, T.; CHATTERJEE, A. A novel binary adaptive weight gsa based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. *Engineering Applications of Artificial Intelligence*, v. 33, p. 80–90, 2014.
- CHEN, J.; KELLOKUMPU, V.; ZHAO, G.; PIETIKAINEN, M. Rlbp: Robust local binary pattern. In: . [S.l.: s.n.], 2013. p. 122.1–122.11. ISBN 1-901725-49-9.
- CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.
- CRIMINISI, A.; ZISSERMAN, A. Shape from texture: Homogeneity revisited. v. 1, p. 2, 01 2000.
- CROSS, G. R.; JAIN, A. K. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, p. 25–39, 1983.
- DANA, K.; GINNEKEN, B. V.; NAYAR, S.; KOENDERINK, J. Curet: Columbia-utrecht reflectance and texture database. 1997. Disponível em: <<http://www1.cs.columbia.edu/CAVE/software/curet/>>.
- DANA, K. J.; GINNEKEN, B. van; NAYAR, S. K.; KOENDERINK, J. J. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, v. 18, p. 1–34, 1999.
- DERIN, H.; ELLIOTT, H.; CRISTI, R.; GEMAN, D. Bayes smoothing algorithms for segmentation of binary images modeled by markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, p. 707–720, 1984.

- FATHI, A.; NAGHSH-NILCHI, A. R. Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern Recognition Letters*, v. 33, p. 1093–1100, 2012.
- GHOLAMI, R.; FAKHARI, N. Support vector machine: Principles, parameters, and applications. *Handbook of Neural Computation*, p. 515–535, 2017.
- GUO, Y.; ZHAO, G.; PIETIKÄINEN, M. Discriminative features for texture description. *Pattern Recognition*, Elsevier, v. 45, n. 10, p. 3834–3843, 2012.
- GUO, Z.; WANG, X.; ZHOU, J.; YOU, J. Robust texture image representation by scale selective local binary patterns. *IEEE Transactions on Image Processing*, IEEE, v. 25, n. 2, p. 687–699, 2015.
- GUO, Z.; ZHANG, L.; ZHANG, D. Rotation invariant texture classification using lbp variance (lbpv) with global matching. *Pattern recognition*, Elsevier, v. 43, n. 3, p. 706–719, 2010.
- HAFIANE, A.; PALANIAPPAN, K.; SEETHARAMAN, G. Joint adaptive median binary patterns for texture classification. *Pattern Recognition*, Elsevier, v. 48, n. 8, p. 2609–2620, 2015.
- HARALICK, R. M. et al. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, 1979.
- HAYMAN, E.; CAPUTO, B.; FRITZ, M.; EKLUNDH, J.-O. On the significance of real-world conditions for material classification. In: *European conference on computer vision*. [S.l.]: Springer, 2004. v. 3024, p. 253–266.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.
- HERSEY, I.; BRODATZ, P. Textures: A photographic album for artists and designers. *Leonardo*, v. 1, p. 91, 1968.
- HONG, X.; ZHAO, G.; PIETIKAINEN, M.; CHEN, X. Combining lbp difference and feature correlation for texture description. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 23, p. 2557–68, 2014.
- HUANG, D.; SHAN, C.; ARDABILIAN, M.; WANG, Y.; CHEN, L. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 41, p. 765–781, 2011.
- HUGHES, D.; CORRELL, N. Texture recognition and localization in amorphous robotic skin. *Bioinspiration & biomimetics*, IOP Publishing, v. 10, n. 5, p. 055002, 2015.
- IAKOVIDIS, D. K.; KERAMIDAS, E. G.; MAROULIS, D. Fuzzy local binary patterns for ultrasound texture characterization. In: *International conference image analysis and recognition*. [S.l.]: Springer, 2008. p. 750–759.
- JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, v. 22, n. 1, p. 4–37, 2000.

- JIN, H.; LIU, Q.; LU, H.; TONG, X. Face detection using improved lbp under bayesian framework. In: *Third International Conference on Image and Graphics (ICIG'04)*. [S.l.]: IEEE, 2004. p. 306–309. ISBN 0-7695-2244-0.
- JING, J.; ZHANG, H.; WANG, J.; LI, P.; JIA, J. Fabric defect detection using gabor filters and defect classification based on lbp and tamura method. *Journal of The Textile Institute - J TEXT INST*, v. 104, p. 1–10, 01 2012.
- JULESZ, B. Visual pattern discrimination. *IRE transactions on Information Theory*, IEEE, v. 8, n. 2, p. 84–92, 1962.
- JULESZ, B. Textons, the elements of texture perception, and their interactions. *Nature*, Nature Publishing Group, v. 290, n. 5802, p. 91, 1981.
- KANNALA, J.; RAHTU, E. Bsif: Binarized statistical image features. In: . [S.l.: s.n.], 2012. p. 1363–1366. ISBN 978-1-4673-2216-4.
- LANDY, M. S.; GRAHAM, N. Visual perception of texture. *The visual neurosciences*, MIT press, p. 1106, 2004.
- LAZEBNIK, C. S. S.; 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.
- LEWIS, J.-P. Texture synthesis for digital painting. In: *ACM SIGGRAPH Computer Graphics*. [S.l.]: ACM, 1984. v. 18, n. 3, p. 245–252.
- LIAO, S.; LAW, M. W. K.; CHUNG, A. C. S. Dominant local binary patterns for texture classification. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 18, p. 1107–18, 2009.
- 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.
- LIU, L.; LAO, S.; FIEGUTH, P. W.; GUO, Y.; WANG, X.; PIETIKÄINEN, M. Median robust extended local binary pattern for texture classification. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 25, p. 1368–81, 2016.
- LIU, L.; LONG, Y.; FIEGUTH, P. W.; LAO, S.; ZHAO, G. Brint: binary rotation invariant and noise tolerant texture classification. *IEEE transactions on Image Processing*, IEEE, v. 23, n. 7, p. 3071–3084, 2014.
- LIU, L.; ZHAO, L.; LONG, Y.; KUANG, G.; FIEGUTH, P. Extended local binary patterns for texture classification. *Image and Vision Computing*, Elsevier, v. 30, n. 2, p. 86–99, 2012.
- LOBAY, A.; FORSYTH, D. A. Shape from texture without boundaries. *International Journal of Computer Vision*, Springer, v. 67, n. 1, p. 71–91, 2006.
- MAANI, R.; KALRA, S.; YANG, Y.-H. Rotation invariant local frequency descriptors for texture classification. *IEEE Transactions on Image Processing*, v. 22, p. 2409–2419, 2013.

- MAATEN, L. v. d.; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. Nov, p. 2579–2605, 2008.
- MALLIKARJUNA, P.; TARGHI, A. T.; FRITZ, M.; HAYMAN, E.; CAPUTO, B.; EKLUNDH, J.-O. The kth-tips2 database. *KTH Royal Institute of Technology*, 07 2006.
- MATERKA, A.; STRZELECKI, M. et al. Texture analysis methods—a review. *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, p. 9–11, 01 1998.
- MIR, A.; HANMANDLU, M.; TANDON, S. Texture analysis of ct images. *IEEE Engineering in Medicine and Biology Magazine*, IEEE, v. 14, n. 6, p. 781–786, 1995.
- MIRMEHDI, M.; XIE, X.; SURI, J. Handbook of texture analysis. Imperial College Press, London, UK, UK, 2009.
- NAILON, W. H. Texture analysis methods for medical image characterisation. *Biomedical Imaging*, INTECH Open Access Publisher London, 2010.
- NANNI, L.; LUMINI, A.; BRAHNAM, S. Local binary patterns variants as texture descriptors for medical image analysis. *Artificial Intelligence in Medicine*, v. 49, p. 117–125, 2010.
- OJALA, T.; MAENPAA, T.; PIETIKAINEN, M.; VIERTOLA, J.; KYLLONEN, J.; HUOVINEN, S. Outex-new framework for empirical evaluation of texture analysis algorithms. In: *Object recognition supported by user interaction for service robots*. [S.l.]: IEEE, 2002. v. 1, p. 701–706.
- OJALA, T.; PIETIKAINEN, M. Nonparametric multichannel texture description with simple spatial operators. In: *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*. [S.l.]: IEEE, 1998. v. 2, p. 1052–1056. ISBN 0-8186-8512-3.
- OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, Elsevier, v. 29, n. 1, p. 51–59, 1996.
- PIETIKÄINEN, M.; HADID, A.; ZHAO, G.; AHONEN, T. Computer vision using local binary patterns. Springer Science & Business Media, v. 40, 2011.
- QI, X.; QIAO, Y.; LI, C.-G.; GUO, J. Multi-scale joint encoding of local binary patterns for texture and material classification. In: *BMVC*. [S.l.: s.n.], 2013. p. 1–11.
- 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, p. 2199–2213, 2014.
- RAHTU, E.; HEIKKILÄ, J.; OJANSIVU, V.; AHONEN, T. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, v. 30, p. 501–512, 2012.
- REN, J.; JIANG, X.; YUAN, J. Noise-resistant local binary pattern with an embedded error-correction mechanism. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 22, 06 2013.

- SCHAEFER, G.; DOSHI, N. Multi-dimensional local binary pattern descriptors for improved texture analysis. In: . [S.l.: s.n.], 2012. p. 2500–2503. ISBN 978-1-4673-2216-4.
- SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- SCHWARTZ, G.; NISHINO, K. Recognizing material properties from images. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, p. 1–1, 2019.
- TAKALA, V.; PIETIKAINEN, M. Multi-object tracking using color, texture and motion. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2007. p. 1–7.
- TAN, X.; TRIGGS, B. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, IEEE, v. 19, n. 6, p. 1635–1650, 2010.
- TUCERYAN, M.; JAIN, A. K. Texture analysis. In: *Handbook of pattern recognition and computer vision*. [S.l.]: World Scientific, 1993. p. 235–276.
- WANG, Z.; SHEIKH, H. R.; BOVIK, A. C. No-reference perceptual quality assessment of jpeg compressed images. In: *Proceedings. International Conference on Image Processing*. [S.l.]: IEEE, 2002. v. 1, p. I–I.
- WESZKA, J. S.; DYER, C.; ROSENFELD, A. A comparative study of texture measures for terrain classification. *Systems, Man and Cybernetics, IEEE Transactions on*, v. 6, p. 269 – 285, 05 1976.
- XU, Y.; JI, H.; FERMÜLLER, C. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, v. 83, p. 85–100, 2009.
- ZHANG, J.; LIANG, J.; ZHAO, H. Local energy pattern for texture classification using self-adaptive quantization thresholds. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 22, 08 2012.
- ZHANG, J.; TAN, T. Brief review of invariant texture analysis methods. *Pattern Recognition*, v. 35, p. 735–747, 2002.
- ZHAO, G.; AHONEN, T.; MATAS, J.; PIETIKAINEN, M. Rotation-invariant image and video description with local binary pattern features. *IEEE Transactions on Image Processing*, v. 21, p. 1465–1477, 2012.
- ZHAO, Y.; HUANG, D.-S.; JIA, W. Completed local binary count for rotation invariant texture classification. *IEEE Transactions on Image Processing*, v. 21, p. 4492–4497, 2012.
- ZHOU, H.; WANG, R.; WANG, C. A novel extended local-binary-pattern operator for texture analysis. *Information Sciences*, v. 178, p. 4314–4325, 2008.
- ZHU, Y.; MAGEE, D.; RATNALINGAM, R.; KESSEL, D. A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures. In: *Proceedings of Medical Image Understanding and Analysis*. [S.l.: s.n.], 2006. p. 61–65.