UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Aprendizado de representações e caracterização de redes complexas com aplicações em visão computacional

Lucas Correia Ribas

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Lucas Correia Ribas

Aprendizado de representações e caracterização de redes complexas com aplicações em visão computacional

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Odemir Martinez Bruno

USP – São Carlos
Janeiro de 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C824a Correia Ribas, Lucas
 Aprendizado de representações e caracterização de
 redes complexas com aplicações em visão
 computacional / Lucas Correia Ribas; orientador
 Odemir Martinez Bruno. -- São Carlos, 2022.
 228 p.

 Tese (Doutorado - Programa de Pós-Graduação em
 Ciências de Computação e Matemática Computacional) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2022.

 1. Reconhecimento de Padrões. 2. Redes
 Complexas. 3. Redes Neurais Artificiais. 4. Visão
 Computacional. 5. Aprendizado de Máquina. I.
 Martinez Bruno, Odemir, orient. II. Título.

Lucas Correia Ribas

**Representation learning and characterization of complex
networks with applications in computer vision**

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Odemir Martinez Bruno

**USP – São Carlos
January 2022**

Aos meus pais, Maria e Luis.

AGRADECIMENTOS

Primeiramente, a Deus, pela minha vida e por tudo que tens feito nela.

Aos meus pais, Luis e Maria, pela confiança, orações, apoio incondicional e carinho que manifestaram ao longo da minha vida. Há mais de 10 anos, eu resolvi sair de casa em busca de um sonho que foi crescendo... e, ao longo destes anos, vocês foram a pedra fundamental que me deu sustentação. Apesar da distância e incertezas, a cada novo desafio, o apoio incondicional e força de vocês me manteve firme nesta caminhada. Por isso, mais uma vez, afirmo: sem vocês, nada disso seria possível!

Ao meu irmão, Thiago, e cunhada, Jhenifer, pelo apoio incondicional e por sempre estar ao meu lado.

À Fabiana, pelos muitos momentos, companheirismo, carinho e amor demonstrado ao longo destes anos.

Ao meu orientador Odemir Martinez Bruno, pela confiança, orientação, amizade e participação ativa na minha formação acadêmica. Por sempre me incentivar e acreditar no potencial dos seus alunos, mesmo quando desacreditados.

Ao professor Antoine Manzanera da École Nationale Supérieure de Techniques Avancées (ENSTA/IP de Paris), pela supervisão, discussões e contribuições para com o trabalho durante o estágio de pesquisa na ENSTA.

Aos amigos do *Scientific Computing Group* e da universidade pelos momentos de descontração, discussão e troca de conhecimento.

Aos professores Amaury A. de Castro Jr e Wesley Nunes Gonçalves por me incentivarem desde a graduação e pelas contribuições na minha formação.

Ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC/USP), pela oportunidade, aos professores e funcionários e a todos, que direta ou indiretamente, contribuíram para com este trabalho. Ao pessoal da cantina do ICMC, em especial a Aurinha, pelos almoços fora de hora, café e momentos de descontração.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo número 2016/23763-8 pela concessão de bolsa no país; à FAPESP, processo número 2019/03277-0 pela concessão de bolsa no exterior, que possibilitaram o desenvolvimento desta tese.

As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade do(s) autor(es) e não necessariamente refletem a visão da FAPESP.

RESUMO

RIBAS, L. C. **Aprendizado de representações e caracterização de redes complexas com aplicações em visão computacional**. 2022. 228 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

As redes complexas têm sido utilizadas como ferramenta de estudo em diversas áreas da ciência devido ao seu carácter multidisciplinar e perspectiva inovadora em relação à análise de dados tradicional. Em visão computacional, diversas abordagens baseadas em redes complexas foram propostas ao longo dos últimos anos com resultados promissores. Tal sucesso é explicado pela capacidade de modelar e quantificar a complexidade presente em muitas imagens, principalmente das advindas da natureza que possui um comportamento não linear. Embora os resultados sejam promissores, técnicas de modelagem otimizadas e métodos de caracterização mais robustos para descrever a complexidade da topologia das redes são necessários. Este trabalho tem como objetivo investigar e propor novas técnicas de modelagem e caracterização de redes complexas focando na aplicação em problemas de visão computacional. Especificamente, foram investigados problemas clássicos de análise de imagens e vídeos: análise de texturas (em níveis de cinza, coloridas e dinâmicas) e formas. Em relação à modelagem, foi estudada uma abordagem eficaz e otimizada para mapear as imagens e vídeos de texturas em redes complexas direcionadas. A respeito da caracterização, com base em métricas clássicas e autômatos celulares, foram propostas duas melhorias: (i) padrões binários em autômatos de rede (LLNA-BP) para caracterização de redes complexas e (ii) redes de transformada da distância para análise de formas. Além disso, foi investigado o aprendizado de representações usando redes neurais randomizadas (RNNs); uma arquitetura simples e com rápido algoritmo de aprendizado. Nesse sentido, como nova forma de caracterização, foram propostas representações formadas pelos pesos da camada de saída de RNNs treinadas com características topológicas de redes complexas. Ao combinar as técnicas desenvolvidas de modelagem e de aprendizado de representações, vários métodos foram propostos para análise de texturas em níveis de cinza, coloridas e dinâmicas e análise de formas. Resultados promissores foram alcançados pelos métodos desenvolvidos em comparação aos métodos da literatura em tarefas de classificação usando vários conjuntos de dados que são referência. Adicionalmente, para avaliar o potencial dos métodos desenvolvidos, foram investigadas cinco aplicações em problemas reais nas áreas de biologia, botânica, físico-química e medicina, alcançando resultados interessantes e contribuindo para o desenvolvimento destas áreas.

Palavras-chave: Reconhecimento de Padrões, Redes Complexas, Aprendizado de Máquina, Redes Neurais Artificiais, Visão Computacional.

ABSTRACT

RIBAS, L. C. **Representation learning and characterization of complex networks with applications in computer vision**. 2022. 228 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Complex networks have been used as a study tool in several areas of science due to their multidisciplinary character and innovative perspective in relation to traditional data analysis. In computer vision, several approaches based on complex networks have been proposed over the last few years with promising results. Such success is explained by the ability to model and quantify the complexity present in many images, especially those arising from nature, which has a non-linear behavior. Although the results are promising, optimized modeling techniques and more robust characterization methods to describe the complexity of the network topology are needed. This work aims to investigate and propose new modeling and characterization techniques for complex networks focusing on application to computer vision problems. Specifically, two classic problems of image and video analysis were investigated: texture (gray-level, color and dynamics) and shape analysis. Regarding modeling, an efficient and optimized approach to map texture images and videos into directed complex networks was studied. Regarding the characterization, based on classical metrics and cellular automata, two improvements were proposed: (i) binary patterns in network automata (LLNA-BP) for characterization of complex networks and (ii) distance transform networks for shape analysis. Furthermore, the learning of representations using randomized neural networks (RNNs) was investigated; a simple architecture and fast learning algorithm. In this sense, as a new form of characterization, representations formed by the weights of the output layer of RNNs trained with topological characteristics of complex networks were proposed. By combining the developed techniques of modeling and learning representations, several methods were proposed for analyzing gray-level, color and dynamic textures and shape analysis. Promising results have been achieved by the developed methods in comparison to literature methods in classification tasks using several benchmark datasets. Additionally, to assess the potential of the developed methods, five applications were investigated in real problems in the areas of biology, botany, physical-chemistry and medicine, achieving interesting results and contributing to the development of these areas.

Keywords: Pattern Recognition, Complex Networks, Machine Learning, Artificial Neural Networks, Computer Vision.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão geral do desenvolvimento desta tese. As setas representam o fluxo metodológico para o desenvolvimento dos métodos, em que cada cor representa um tipo de abordagem.	33
Figura 2 – Construção do autômato de rede Life-Like usando a regra B25-S8 para uma rede com $N = 21$ evoluída durante três iterações. No topo, da esquerda para a direita, a evolução do LLNA é visualizada. Os estados são representados por branco (vivo) e preto (morto). Na parte inferior, detalha-se passo a passo a função de transição B25-S8.	42
Figura 3 – Ilustração de modelagens em redes complexas. (a) Textura estática, (b) contorno de forma, (c) textura dinâmica. Os pixels são representados por vértices e as arestas conectam os vértices de acordo com um raio de distância e similaridade de cor.	48
Figura 4 – Esquema do modelo de neurônio perceptron proposto por Rosenblatt (1958).	50
Figura 5 – Esquema de uma rede neural randomizada com uma camada oculta.	52
Figura 6 – Organização das camadas da arquitetura AlexNet.	57
Figura 7 – Exemplo de vizinhança completa para três raios diferentes. Os pixels representados por esferas em verde serão conectados com o pixel central colorido em azul.	64
Figura 8 – Estrutura de redes complexas direcionadas modelando pixels com diferentes valores do parâmetro de modelado r	64
Figura 9 – Vizinhança radialmente simétrica para três raios diferentes considerando apenas um canal. Os pixels representados por esferas em verde serão conectados com o pixel central colorido em azul.	65
Figura 10 – Estruturas multi-camadas das SSNs que modelam três canais de uma imagem 3×3 com raio $r = 1$	67
Figura 11 – Modelagem de três quadros em redes complexas direcionadas usando raio $r = 1$	69

Figura 12 – Diagrama geral do processo de aprendizado de representações de redes complexas empregado nos métodos desenvolvidos neste capítulo. (a) Uma única amostra é modelada como rede complexa e, para cada vértice, um vetor de atributos é construído e um rótulo é atribuído. (b) Todos os vetores de atributos dos vértices da rede complexa formam a matriz de entrada que treinará a RNN. (c) Os pesos da camada de saída da RNN treinada serão utilizados para compor a representação final em (d).	72
Figura 13 – Construção de um vetor de atributos de entrada e rótulo correspondente para o grau de saída $\bar{k}_{v_i}$ usando diferentes valores de r para modelar as redes complexas. (a) O grau de saída $\bar{k}_{v_i}$ do vértice v_i para diferentes raios r é usado para construir o vetor de entrada e a intensidade de cinza do pixel i é usada como rótulo de saída. (b) A RNN é treinada usando matriz $X_{\bar{k}}$ e Y construídos considerando todos os vértices, e os pesos treinados da camada de saída são usados como assinatura.	76
Figura 14 – Acurácias para a assinatura $\vec{\Upsilon}(Q)_R$ usando diferentes valores de Q nas quatro bases de dados.	77
Figura 15 – Acurácias usando o vetor $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ para os dois melhores conjuntos de Q com diferentes valores de raio máximo R	79
Figura 16 – Vizinhança de um vértice/pixel.	81
Figura 17 – Esquema do método PCL para construir o vetor de atributos e rótulo a partir de cada vértice.	82
Figura 18 – Acurácias da representação $\vec{\Theta}(Q)_{r_1, r_2, \dots, r_m}$ usando $Q = 4$ para diferentes valores de r nas quatro bases de dados.	83
Figura 19 – Resultados de classificação para a representação $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ com diferentes valores de Q	84
Figura 20 – Uma imagem de textura (a) da base de dados USPTex é modelada como uma rede complexa direcionada para diferentes valores de r e representações visuais das medidas de cada vértice são mostradas (d)-(f). Os valores dos pixels são obtidos normalizando cada medida de forma que o grau máximo do vértice para cada raio (4,60,180 e 364) corresponda à intensidade 255 (branco).	86
Figura 21 – Amostras de textura (a) da base de dados Brodatz e seus respectivos vetores de características $\vec{f}_k$ em (b). Este vetor de característica foi calculado usando os parâmetros $R = 04$ e $Q = 04$	87
Figura 22 – Vetores de pesos da camada de saída gerados pela abordagem PLCNN de amostras de textura de várias classes. Cada amostra é representada nas colunas e as linhas são os pesos.	88

Figura 23 – Exemplos de uma textura da base Brodatz (BRODATZ, 1966) rotacionadas em 11 diferentes ângulos. Da esquerda para a direita, de cima para baixo: 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°.	91
Figura 24 – Representação visual da SSN ($R = 8$); imagens resultantes são obtidas pelo cálculo de medidas de RC para um vértice em cada camada da rede, para depois ser normalizada em pixels RGB de 8-bits. G_S , G e G_T referem-se aos grafos espacial, completo e espectral, respectivamente.	96
Figura 25 – Ilustração da criação da matriz de atributos de entrada $X_{\bar{k}}^G$ com o grau de saída da versão completa G . A imagem da textura colorida é modelada em redes complexas com diferentes valores de r (a figura mostra um exemplo para $r = 1$). O grau de saída $\bar{k}_{v_i}$ dos vértices para diferentes valores de r são usados para criar a matriz de entrada $X_{\bar{k}}^G$ e seu grau de saída $\bar{k}_{v_i}^R$ para raio R é usado como rótulo.	97
Figura 26 – Esquema geral do método proposto SSRNN para uma imagem colorida e valores de parâmetros R e Q . Primeiramente, a imagem colorida é representada pela modelagem SSN direcionada, resultando em três RCs: G , G_S e G_T . Observe que nessa etapa o raio de conexão varia de $\{1, 2, \dots, R - 1\}$. Em seguida, as matrizes de atributos de entrada X (usando uma das medidas $\bar{k}$, $\bar{s}$ e $\hat{s}$) são criadas para as três RCs com base no conjunto de raios e então as RNNs são treinadas para prever o conjunto Y formado pelos graus ($\bar{k}$ ou $\hat{k}$) para o raio R . Por fim, os vetores da camada de saída com os pesos aprendidos compõem o vetor de características.	98
Figura 27 – Acurácia média nas três bases de dados para diferentes valores de R e suas combinações com $Q = 04$.	99
Figura 28 – Acurácias para o método SSRNN para diferentes valores de Q (diagonal principal) e suas combinações.	100
Figura 29 – Principais etapas do método proposto RNNDT: (1) conjunto de matrizes de vetores de entrada e rótulos para cada plano ortogonal; (2) treinamento da rede neural randomizada; (3) os pesos médios da camada de saída para cada plano ortogonal são concatenados.	106
Figura 30 – Exemplo de extração de um vetor de atributos de entrada e seu respectivo rótulo de uma janela de uma fatia do plano ortogonal XY para treinar a RNN.	108
Figura 31 – Construção do vetor de entrada e rótulo de uma RC espacial modelando texturas dinâmicas com diferentes valores de r e usando o grau de saída k_i .	112
Figura 32 – Fluxograma da abordagem proposta CPNN para análise de textura dinâmica.	113
Figura 33 – Acurácia média para diferentes valores e combinação de um ou dois raios nas bases UCLA-50 e Dyntex++.	114

Figura 34 – Resultados de classificação para a representação $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ com um único ($m = 1$) ou dois diferentes ($m = 2$) valores de Q nas bases de dados Dyntex ++ e UCLA-50. As células diagonais correspondem a valores únicos de Q , enquanto as demais se referem as combinações de dois valores distintos de Q .	115
Figura 35 – Sumarização do método proposto DTN de caracterização de formas baseado em transformada da distância e redes complexas.	124
Figura 36 – Exemplos de modelagens de redes: (a) diferentes raios de dilatação r modelados como redes complexas; (b) raio de dilatação $r = 0$ modelado com uma rede transformada por diferentes valores de limiar t	127
Figura 37 – Características de grau de três raios de dilatação de um mesmo contorno. . .	128
Figura 38 – Resultado de reconhecimento de formas para três bases de dados como uma função do raio inicial r_0 e o raio final r_f	130
Figura 39 – Modelagem de um contorno de folha como redes complexas usando três valores diferentes de limiar t . Os pontos de contorno estão em preto e as arestas em cinza.	132
Figura 40 – Ilustração da criação da matriz de entrada X_D^3 e vetor de rótulo Y^3 a partir do TFM de grau $\mathbf{F_K}$ e usando $\eta = 3$. Considerando a linha 1, os valores nas três primeiras colunas formam o primeiro vetor de entrada (caixa preta) em X_Y^3 e a coluna 4 o seu respectivo rótulo no vetor Y^3 (círculo preto). O segundo vetor de entrada é composto de valores entre as colunas 2 e 4 (caixa vermelha) com o seu respectivo rótulo como a coluna 5 (círculo vermelho). Esse processo é repetido para todas as linhas e colunas.	134
Figura 41 – Esquema geral da proposta CRSA considerando um único valor de Q e η . (a)-(b) Um contorno de forma é modelado em RCs, que são transformadas por diferentes valores de limiar. (c) Os mapas de características topológicas são calculados a partir da evolução da RC. (d) Matrizes de entrada são obtidas dos TFM e em (e) usadas para treinar as RNNs. Finalmente, os pesos de saída aprendidos formam a representação da forma $\vec{\phi}(Q)_\eta$	136
Figura 42 – Resultados para os vetores $\vec{\phi}(Q)_\eta$ e $\vec{\psi}(Q)_{\eta_1, \eta_2}$ usando diferentes valores de η .	137
Figura 43 – Acurácias para o vetor $\vec{Y}_{Q_1, Q_2, \dots, Q_n}$ com combinações de valores de Q_1 e Q_2 .	138
Figura 44 – Acurácias para diferentes valores de limiar de incremento t_{inc} nas bases ETH-80 e Folhas Suecas. As maiores acurácias são obtidas com $t_{inc} = 0,005$, fornecendo 96,88% e 94,08% nas bases Folhas Suecas e ETH-80, respectivamente.	139
Figura 45 – Formas geométricas modeladas como redes complexas (vértices em preto e arestas em cinza) usando três valores de limiar diferentes. Cada linha representa uma forma e as colunas são os valores de limiar de t_1 , t_2 e t_3 , em que t_1 é o menor valor, t_2 um valor intermediário e t_3 o maior.	141

- Figura 46 – Pesos aprendidos da última camada da RNN na abordagem CRSA usando janela $\eta = 7$ e $Q = 19$. Cada linha corresponde ao peso de uma conexão entre a camada oculta e a camada de saída, ao passo que as colunas representam as formas geométricas. O valor do peso aprendido normalizado entre $[0, 1]$ é representado pela escala de cinza. 142
- Figura 47 – Exemplos de pesos aprendidos usando a abordagem CRSA com janela $\eta = 7$ e $Q = 19$. Cada linha corresponde ao peso de uma conexão entre a camada oculta e a camada de saída, ao passo que as colunas representam as formas de folhas de 4 espécies (2 folhas por espécie). O valor do peso aprendido normalizado entre $[0, 1]$ é representado pela escala de cinza. 143
- Figura 48 – Exemplo de formas de folhas da base Folhas USP em diferentes condições. (a) contornos rotacionados, (b) contornos escalados e (c) contornos com ruído. 148
- Figura 49 – Esquema de reconhecimento de padrões proposto baseado em padrões binários extraídos dos padrões de evolução temporal do LLNA (LLNA-BP). (a) Considere um problema em que as amostras são modeladas como rede complexa. (b) Para cada rede, em (c) o LLNA evolui com uma regra previamente selecionada com os estados representados por branco (vivo ou um) e preto (morto ou zero). De cima para baixo, o padrão de evolução temporal produzido pela dinâmica LLNA da rede dada. (d) Sequências de zeros e uns são concatenadas para produzir padrões binários de tamanhos diferentes, e.g., $D = 3, D = 4, \dots D_n$, a partir do qual histogramas são obtidos, respectivamente, (e) formando, assim, o vector de características utilizando várias estratégias, (e.g., concatenando as distribuições de frequência global $\vec{Y}_{D_1, D_2, \dots, D_n}^g$), que são então usados para (f) a tarefa de classificação de redes. 155
- Figura 50 – Duas amostras de padrões de evolução temporal para quatro modelos de redes diferentes: aleatório, pequeno mundo, livre de escala e geográfico, e seus histogramas. (a) Os TEPs correspondem a redes com $N = 500$ vértices, grau médio $\langle k \rangle = 10$, sendo evoluídos com a regra B135678-S03456 durante $T = 350$ passos de tempo. Cada coluna é um modelo de rede diferente e cada linha é uma amostra de rede diferente. (b) Histogramas correspondentes ao histograma global $H_g^D(p)$ considerando diferentes valores de comprimento do padrão binário D . Cada curva corresponde a uma das amostras em (a). . . 158
- Figura 51 – Exemplos de imagens da família *Ariidae*. Cada linha representa uma espécie e as colunas são amostras da mesma espécie. 171
- Figura 52 – Matriz de confusão para o resultado do método CRSA usando o classificador SVM (92,86%) no experimento com todas as famílias. (a) Amostra da espécie *Cryptarius truncatus* classificada incorretamente como da espécie *Nemapteryx caelata*. (b) Exemplo de espécime da classe *Nemapteryx caelata* usada no treinamento. 173

Figura 53 – Exemplos de imagens de Jacaranda Caroba. Da esquerda para direita e de cima para baixo as amostras são de 5mg/L, 15mg/L, 30mg/L, 60mg/L, 1162mg/L e Controle.	174
Figura 54 – Matriz de confusão para a classificação de níveis de poluente em imagens de Jacaranda Caroba. Essa matriz é referente ao melhor resultado de 95,3% obtido com o método proposto SSRNN ¹ usando LOO e com a imagem RGB.	178
Figura 55 – Exemplos de imagens de cada classe capturadas com SEM de unidades sensoriais.	179
Figura 56 – Projeção t-SNE em 2D das assinaturas calculadas pelos métodos CNRNN, AHP, GLDM e LCFNN. Cada ponto representa uma amostra e a cor indica a classe para fins de visualização.	181
Figura 57 – Osso segmentado em que os pontos vermelhos representam os marcadores anatômicos e a linha tracejada representa a borda tibial.	183
Figura 58 – Exemplos de imagens ROI para K&L grau 0 (CC) na primeira linha e K&L grau 2 (OP) na segunda linha.	184
Figura 59 – Exemplos de 4 amostras (colunas) de 7 classes (linhas) da base de dados de folha.	187
Figura 60 – Exemplos de imagens de textura de várias classes da base de dados Brodatz.	218
Figura 61 – Amostras de textura de diferentes classes da base de dados Outex.	219
Figura 62 – Imagens de textura coloridas de 30 classes da base de dados USPTex.	220
Figura 63 – Amostras de textura coloridas de classes da base Vistex.	221
Figura 64 – Exemplos de textura de diversas classes da base MBT.	222
Figura 65 – Primeiro quadro de amostras de textura dinâmica de 30 classes da base de dados Dyntex++.	223
Figura 66 – Exemplos do primeiro quadro de amostras de textura dinâmica de diversas classes da base de dados UCLA.	224
Figura 67 – Amostas das 9 classes de forma da base de Kimia.	225
Figura 68 – Exemplos de cada classe de forma da base de dados ETH-80.	226
Figura 69 – Amostras de contorno de folhas da base de dados Folhas USP.	227

LISTA DE TABELAS

Tabela 1	– Resultados de classificação para o vetor $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ usando diferentes valores de Q e suas combinações para o raio máximo $R = 4$	78
Tabela 2	– Resultados de classificação considerando diferentes combinações de raios R usando $Q = \{4, 9, 14\}$	80
Tabela 3	– Acurácias usando diferentes conjuntos de raio R com $Q = \{4, 14, 19\}$	80
Tabela 4	– Resultados para o vetor $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ combinando três valores diferentes de Q	85
Tabela 5	– Comparação das acurácias de diferentes métodos de análise de textura em quatro bases de dados usando o classificador LDA. Esses resultados foram reproduzidos para esta tese.	90
Tabela 6	– Comparação de acurácias de diferentes métodos de análise de textura em bases de dados rotacionadas. Um subconjunto dos resultados comparados está presente em (JUNIOR; BACKES, 2016).	92
Tabela 7	– Comparação do tempo de processamento dos métodos de análise de texturas em níveis de cinza para extração de assinatura de uma imagem.	93
Tabela 8	– Resultados para o vetor de características $\vec{\Psi}(04, 10)_{Q_1, Q_2, \dots, Q_m}$ combinando $\{Q_1, Q_2, Q_3\}$ para os raios $R = \{4, 10\}$	101
Tabela 9	– Resultados de comparação para o método proposto e métodos da literatura usando o classificador LDA (<i>leave-one-out</i>) em três bases de dados de texturas coloridas. Resultados do método proposto com * foram calculados com validação cruzada em 10 pastas. As acurácias do método RNN-RGB foram extraídas diretamente do artigo original, em que o resultado para a base MBT não está disponível.	103
Tabela 10	– Resultados de classificação (RNNDT) para o vetor λ_Q com diferentes valores de Q	110
Tabela 11	– Comparação do método proposto RNNDT com vetor de característica Y combinando diferentes valores de Q	110
Tabela 12	– Acurácia para o vetor $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ usando três valores de Q	115
Tabela 13	– Tempo de processamento computacional em segundos dos métodos propostos e outros métodos para calcular o vetor de características.	116

Tabela 14 – Comparação na base de dados UCLA-50 (classificador 1NN e validação cruzada com 4 pastas). Para os métodos com '*', os resultados são retirados de (TIWARI; TYAGI, 2016a), (TIWARI; TYAGI, 2017) e (GONÇALVES; MACHADO; BRUNO, 2015), os resultados com + são obtidos do artigo original, enquanto os demais foram gerados.	117
Tabela 15 – Comparação dos métodos propostos com outros métodos de textura dinâmica nas bases UCLA de 9 e 8 classes (classificador 1NN e metade das amostras para treinamento e o restante para teste). Os resultados com '*' são retirados de (TIWARI; TYAGI, 2016a) e (TIWARI; TYAGI, 2017), os resultados com + são obtidos do artigo original, enquanto os demais foram gerados.	118
Tabela 16 – Comparação do método proposto com outros na base de dados Dyntex++ (classificador 1NN e validação cruzada com 10 pastas). Os resultados com + são retirados do artigo original, enquanto os outros foram gerados.	119
Tabela 17 – Resultados do método proposto para diferentes conjuntos de limiares T e usando raio de dilatação $r = 0$	129
Tabela 18 – Resultados de classificação para a representação $\vec{Y}_{Q_1, Q_2, \dots, Q_n}$ calculada com diferentes valores de Q	138
Tabela 19 – Resultados comparativos entre duas bases de dados formas genéricas usando os classificadores LDA e SVM.	146
Tabela 20 – Acurácias das duas bases de dados de formas de folhas para os classificadores LDA e SVM.	147
Tabela 21 – Resultados para a versão rotacionada e escalada da base de dados Folhas USP.	148
Tabela 22 – Comparação dos métodos propostos com outros da literatura para a base de dados Folhas USP corrompida por diferentes níveis de ruído usando o classificador LDA.	149
Tabela 23 – Resultados para a base de dados de rede social usando diferentes vetores de características e combinações de valores de D	164
Tabela 24 – Resultados para diferentes combinações de valores de D com o vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$ em bases de dados sintéticas. Desvios padrão entre $\pm 0,0$ e $\pm 0,09$ não são mostrados.	165
Tabela 25 – Acurácia (%) e desvio padrão para a classificação das três base de dados de metabólitos usando diferentes combinações de valores de D com o vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$	166
Tabela 26 – Comparação da acurácia (%) e desvio padrão obtidos pela abordagem proposta, o LLNA tradicional e as medidas de rede aplicadas na classificação de 14 bases de dados de rede. O LLNA-BP usou o vetor de características $\vec{\Psi}_{5,7}$. O LLNA usou $[\vec{\mu}_S, \vec{\mu}_W, \vec{\mu}_L]$ como vetor de características, enquanto o vetor das medidas clássicas de rede foi $[\langle k \rangle, \langle H_{k_1} \rangle, \langle H_{k_2} \rangle, \langle cc \rangle, l, \rho_P]$	167

Tabela 27 – Resultados comparativos para a classificação de espécies de peixe através do contorno do otólito usando os classificadores LDA e SVM. Quatro experimentos foram realizados, sendo testes com cada família separada e também com todas as famílias juntas.	170
Tabela 28 – Resultado de classificação para diferentes métodos na classificação de níveis de poluente em imagens de Jacaranda Caroba em formatos em escala de cinza e RGB, e com diferentes tamanhos. Os resultados foram obtidos com o classificador LDA, enquanto para os métodos CNNs que foram treinados, os resultados são dados pelo próprio classificador da rede neural.	176
Tabela 29 – Descrição do número de amostras da base de dados com imagens de genossensores.	179
Tabela 30 – Acurácia e desvio padrão (em parênteses) para os experimentos binário e multi-classe da base de dados composta por imagens de biossensores para diagnóstico de câncer de próstata.	180
Tabela 31 – Acurácia para os grupos calculados com k-means nos casos binário e multi-classe na base de dados de genossensores.	182
Tabela 32 – Acurácias de classificação usando SVM para os métodos aplicados na base de OA.	185
Tabela 33 – Acurácias para diferentes métodos de texturas aplicados no problema de classificação de espécies (folhas e janelas) de plantas usando o classificador LDA. Os métodos foram executados nas imagens convertidas para escala de cinza, com exceção dos métodos SSRNN ¹ , SSRNN ² e SSN que usaram a imagem RGB.	188

LISTA DE ABREVIATURAS E SIGLAS

INN	<i>1-Nearest Neighborhood</i>
CA	<i>Cellular Automata</i>
CLBP	<i>Complete Local Binary Patterns</i>
CNN	<i>Convolutional Neural Network</i>
CNTD	<i>Complex Networks Traditional descriptors</i>
CPNN	<i>Complex Patterns learned using Randomized Neural Network</i>
CRSA	<i>Complex Representation for Shape Analysis</i>
DL	<i>Deep Learning</i>
DT	<i>Dynamic Texture</i>
DTN	<i>Distance Transform Network</i>
EDT	<i>Euclidean Distance Transform</i>
ERCNN	<i>Evolução de Redes Complexas com RNN</i>
GAP	<i>Global Average Pooling</i>
GNN	<i>Graph Neural Network</i>
GPU	<i>Graphical Processing Units</i>
ILSVRC	<i>ImageNet Large Scale Visual Recognition Challenge</i>
KNN	<i>K-Nearest Neighbors</i>
LCG	<i>Linear Congruent Generator</i>
LDA	<i>Linear Discriminant Analysis</i>
LLNA	<i>Life-Like Network Automata</i>
LLNA-BP	<i>Life-Like Network Automata descriptor based on Binary Pattern</i>
LPQ	<i>Local Phase Quantization</i>
MCND	<i>Multilayer Complex Network Descriptors</i>
OA	<i>OsteoArtrite</i>
OAI	<i>Osteoarthritis Initiative</i>
PLCNN	<i>Padrões Locais Complexos usando RNN</i>
RC	<i>Rede Complexa</i>
RNAs	<i>Redes Neurais Artificiais</i>
RNN	<i>Randomized Neural Network</i>
RNNDT	<i>Randomized Neural Network for Dynamic Texture</i>
ROI	<i>Region of Interest</i>

SEM	<i>Scanning Electron Microscopy</i>
SSN	<i>Spatio-Spectral Network</i>
SSR	<i>Spatio-Spectral Representation</i>
SSRNN	<i>Spatio-Spectral Representation learned by RNN</i>
SVM	<i>Support Vector Machine</i>
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>
TEPs	<i>Temporal Evolution Patterns</i>
TFM	<i>Topological Feature Map</i>

SUMÁRIO

1	INTRODUÇÃO	29
1.1	Objetivos	31
1.2	Contribuições	32
1.3	Organização	34
2	REDES COMPLEXAS	35
2.1	Definição	36
2.2	Medidas e Características Topológicas	37
2.3	Autômatos de Redes	39
2.3.1	<i>Autômatos de Rede baseados no Life-Like</i>	40
2.4	Redes Complexas em Visão Computacional	43
3	REDES NEURAIIS ARTIFICIAIS	49
3.1	Redes Neurais Randomizadas	50
3.1.1	<i>Complexidade Computacional</i>	53
3.2	Redes Neurais Profundas	53
3.2.1	<i>Conceitos e Principais Modelos de CNN</i>	54
3.2.2	<i>Transferência de Aprendizado</i>	57
3.2.3	<i>Redes Neurais Profundas em Grafos</i>	58
4	MODELAGEM DE TEXTURAS EM REDES COMPLEXAS DIRE- CIONADAS	61
4.1	Modelagem de Texturas em Níveis de Cinza	62
4.1.1	<i>Complexidade Computacional</i>	63
4.2	Modelagem Espaço-Espectral de Texturas Coloridas	64
4.2.1	<i>Complexidade Computacional</i>	66
4.3	Modelagem Espaço-Temporal de Texturas Dinâmicas	67
4.3.1	<i>Complexidade Computacional</i>	69
4.4	Considerações Finais	70
5	APRENDIZADO DE REPRESENTAÇÕES USANDO RNN	71
5.1	Aprendizado de Representações de Redes Complexas para Análise de Texturas em Níveis de Cinza	73

5.1.1	<i>Abordagem 1: Aprendizado de Representações da Evolução de Redes Complexas (ERCNN)</i>	74
5.1.2	<i>Abordagem 2: Aprendizado de Representações usando Padrões Locais Complexos (PLCNN)</i>	80
5.1.3	<i>Análise das Representações Propostas</i>	84
5.1.4	<i>Resultados e Discussões</i>	87
5.2	<i>Aprendizado de Representação Espaço-Espectral para Análise de Texturas Coloridas (SSRNN)</i>	94
5.2.1	<i>Representação Espaço-Espectral</i>	95
5.2.2	<i>Resultados e Discussões</i>	98
5.2.3	<i>Considerações Finais</i>	102
5.3	<i>Redes Complexas e RNNs para Análise de Texturas Dinâmicas</i> . . .	104
5.3.1	<i>Abordagem 1: Aprendizado de Representações de Três Planos Ortogonais (RNNDT)</i>	105
5.3.2	<i>Abordagem 2: Aprendizado de Representações de Redes Complexas Espaço-Temporal (CPNN)</i>	111
5.3.3	<i>Resultados e Discussões</i>	114
5.4	<i>Considerações Finais</i>	120
6	ANÁLISE DE FORMAS COM REDES COMPLEXAS E RNN . . .	121
6.1	Redes de Transformada da Distância para Análise de Formas (DTN)	122
6.1.1	<i>Transformada da Distância Euclidiana</i>	123
6.1.2	<i>Modelando Raios de Dilatação como Redes</i>	124
6.1.3	<i>Assinatura</i>	125
6.1.4	<i>Complexidade Computacional</i>	128
6.1.5	<i>Estudo de Parâmetros</i>	128
6.2	Aprendizado de Representações Complexas por meio de RNNs em Mapas de Características Topológicas (CRSA)	131
6.2.1	<i>Modelando Contorno como Redes Complexas</i>	131
6.2.2	<i>Mapas de Características Topológicas</i>	133
6.2.3	<i>Representações Propostas</i>	133
6.2.4	<i>Complexidade Computacional</i>	136
6.2.5	<i>Estudo de Parâmetros</i>	136
6.2.6	<i>Interpretando as Representações Aprendidas</i>	140
6.3	Experimentos e Resultados	143
6.3.1	<i>Configuração Experimental</i>	143
6.3.2	<i>Classificação de Formas Genéricas</i>	145
6.3.3	<i>Reconhecimento de Formas de Folhas de Plantas</i>	145
6.3.4	<i>Análise de Invariância à Rotação e Escala</i>	146
6.3.5	<i>Análise de Robustez a Ruído</i>	147

6.4	Discussões e Considerações Finais	149
7	PADRÕES BINÁRIOS DE LLNA PARA CARACTERIZAÇÃO E CLASSIFICAÇÃO DE REDES COMPLEXAS	153
7.1	Padrões Binários de TEPs	154
7.1.1	<i>Histograma Global</i>	<i>156</i>
7.1.2	<i>Histograma de Graus</i>	<i>156</i>
7.1.3	<i>Assinatura</i>	<i>157</i>
7.1.4	<i>Complexidade Computacional</i>	<i>159</i>
7.2	Experimentos	159
7.2.1	<i>Bases de dados</i>	<i>159</i>
7.2.2	<i>Configuração do LLNA-BP</i>	<i>161</i>
7.2.3	<i>Configuração de Classificação</i>	<i>162</i>
7.2.4	<i>Métodos para Comparação</i>	<i>162</i>
7.3	Resultados e Discussões	163
7.3.1	<i>Avaliação dos Parâmetros</i>	<i>163</i>
7.3.2	<i>Comparações e Discussões</i>	<i>164</i>
7.3.3	<i>Tempo de Processamento</i>	<i>167</i>
7.4	Considerações Finais	168
8	APLICAÇÕES	169
8.1	Reconhecimento de Espécies de Peixes baseado no Contorno do Otólito	169
8.1.1	<i>Dados</i>	<i>170</i>
8.1.2	<i>Experimentos e Resultados</i>	<i>170</i>
8.2	Análise de Plasticidade Foliar de Jacaranda Caroba como biossensor natural de poluentes	172
8.2.1	<i>Dados</i>	<i>173</i>
8.2.2	<i>Experimentos e Resultados</i>	<i>174</i>
8.3	Diagnóstico de Câncer de Próstata por meio de Imagens de Genossensores	177
8.3.1	<i>Dados</i>	<i>179</i>
8.3.2	<i>Experimentos e Resultados</i>	<i>179</i>
8.4	Deteccção de Osteoartrite Precoce de Joelho por meio de Imagens de raio-X	181
8.4.1	<i>Dados</i>	<i>182</i>
8.4.2	<i>Experimentos e Resultados</i>	<i>183</i>
8.5	Reconhecimento de Espécies de Plantas baseado em Análise de Textura Foliar	185
8.5.1	<i>Dados</i>	<i>186</i>

8.5.2	<i>Experimentos e Resultados</i>	186
8.6	Considerações Finais	189
9	CONCLUSÃO	191
9.1	Trabalhos Futuros	194
9.2	Trabalhos Gerados	194
REFERÊNCIAS		197
APÊNDICE A	RECONHECIMENTO DE PADRÕES E BASES DE DADOS	217
A.1	Bases de dados	217
A.1.1	<i>Texturas</i>	217
A.1.2	<i>Texturas Dinâmicas</i>	219
A.1.3	<i>Formas</i>	220
A.2	Reconhecimento de Padrões	223
A.2.1	<i>Classificadores</i>	223
A.2.2	<i>Esquemas de validação</i>	227
A.2.3	<i>Métricas</i>	228

INTRODUÇÃO

Ao longo de séculos, o paradigma de pesquisa na ciência foi pautado por uma abordagem reducionista dos dados, em que cada disciplina científica era estudada individualmente (MIRANDA; MACHICAO; BRUNO, 2016). Por outro lado, mais recentemente, suportado pelos avanços científicos e tecnológicos, computadores com alto poder de processamento e armazenamento foram desenvolvidos, contribuindo com a explosão na quantidade de dados disponíveis. Como consequência desses avanços, surge uma alta demanda pelo desenvolvimento de abordagens multidisciplinares e integrativas que englobem várias áreas da ciência. Nesse sentido, a ciência das redes complexas tem se destacado como uma alternativa ao modelo reducionista devido ao seu carácter multidisciplinar e capacidade de representar e estudar qualquer estrutura natural, sistemas e dados complexos.

Os fundamentos teóricos da área de redes complexas surgem com base na interseção de conceitos da teoria dos grafos, física, mecânica estatística e ciência da computação. Assim, esse novo campo de pesquisa não somente apenas estende o formalismo de grafos, mas também propõe métricas e técnicas embasadas em propriedades reais de sistemas complexos para estudá-los (CASANOVA, 2013; COSTA *et al.*, 2007). A sua origem pode ser atribuída ao estudo de redes aleatórias realizado por Erdős e Rényi (1960) em meados dos anos 1960. No entanto, trabalhos mais recentes provocaram uma revolução na área, com destaque para as investigações de redes de pequeno mundo por Watts e Strogatz (1998) e redes livre de escala de Barabási e Albert (1999). A partir desses trabalhos, a pesquisa em redes complexas tem alcançado diversas áreas da ciência, incluindo a biologia (BARABÁSI; OLTVAI, 2004), sociologia (NEWMAN; PARK, 2003), engenharia (NASIRUZZAMAN; POTA; MAHMUD, 2011), computação (CANTERO *et al.*, 2018), entre outros. Um dos principais interesses nas pesquisas que envolvem a área é em como modelar o dado ou sistema de interesse em uma rede para, em seguida, realizar uma análise das suas propriedades topológicas por meio de um conjunto de medidas (COSTA *et al.*, 2007). Em relação à modelagem, a questão está em: como a partir de um conjunto de indivíduos ou elementos, estabelecer o relacionamento entre estes, visando construir um conjunto de relações

ou arestas. Em muitos casos, a própria definição dos elementos que serão atribuídos aos vértices é uma tarefa desafiadora, que precisa ser otimizada. No que tange à análise topológica, diversos estudos têm proposto medidas que extraem características da estrutura de redes com a finalidade de compreender e explicar o funcionamento dos sistemas construídos sobre estas, possibilitando o reconhecimento de padrões intrínsecos.

Atualmente, com a introdução da ciência dos dados e da Era de “*Big Data*”, há uma necessidade por métodos de reconhecimento de padrões que lidam com dados formados por relações não-lineares, tornando as redes complexas uma ferramenta importante em diversas áreas que necessitam de reconhecimento de padrões, tais como, visão computacional (SCABINI *et al.*, 2017; RIBAS *et al.*, 2020; MIRANDA; MACHICAO; BRUNO, 2018; RIBAS; NEIVA; BRUNO, 2019), atribuição de autoria (MACHICAO *et al.*, 2018a; MEHRI; DAROONEH; SHARIATI, 2012; AKIMUSHKIN; AMANCIO; OLIVEIRA JR., 2017), reconstrução filogenética (BANERJEE; JOST, 2008), entre outros. Nesse contexto, a ciência das redes complexas como ferramenta para reconhecimento de padrões surge como uma alternativa promissora para ciência dos dados e inteligência artificial. O reconhecimento de padrões baseado em redes complexas visa classificar um conjunto de redes em várias classes e distingui-las de acordo com suas características intrínsecas (BANERJEE; JOST, 2008; MIRANDA; MACHICAO; BRUNO, 2016). Assim, ao invés de focar nas propriedades topológicas de uma rede isolada, o objetivo é a classificação, agrupamento e comparação de diferentes redes, de modo que cada rede é explorada em sua totalidade (XIN; ZHANG; SHAO, 2018).

Em visão computacional, técnicas baseadas em redes complexas têm sido empregadas, com sucesso em diversos problemas, como uma medida de irregularidade e complexidade (BACKEs, 2010). Em particular, excelentes resultados foram obtidos em aplicações com imagens naturais, tais como, identificação de espécies de plantas, análise de plasticidade foliar e análise de aglomeração de nanopartículas (CASANOVA, 2013; GONÇALVES *et al.*, 2016; MACHADO *et al.*, 2017). Tal desempenho pode ser explicado pela sua capacidade de modelar e quantificar a complexidade presente em muitas imagens, principalmente, nas advindas da natureza. Apesar desses resultados promissores, novas técnicas de modelagem mais rápidas e simples que melhorem a captura da complexidade das imagens precisam ser investigadas mais profundamente. Além disso, em termos de caracterização, o desenvolvimento de técnicas mais sofisticadas pode permitir o aprimoramento da descrição topológica das redes complexas em comparação com medidas clássicas mais simples.

Por outro lado, o grande volume de dados, a crescente complexidade dos problemas a serem processados computacionalmente e a elevada capacidade dos recursos computacionais atualmente disponíveis, são fatores que também favorecem a expansão da área de aprendizado de máquina (GAMA *et al.*, 2011). Nessa área, a pesquisa com técnicas de redes neurais artificiais tem atraído crescente interesse da comunidade científica e de empresas. Esse recente interesse foi motivado, principalmente, pelos excelentes resultados dos modelos de redes neurais profundas,

que são ferramentas poderosas empregadas em diferentes problemas, tais como, processamento de áudio (PURWINS *et al.*, 2019), processamento de linguagem natural (YOUNG *et al.*, 2018), visão computacional (SZEGEDY *et al.*, 2017), entre outras. Esses resultados, e o potencial integrativo e multidisciplinar das redes complexas, também motivaram pesquisas para a combinação de ambas as ferramentas para reconhecimento de padrões (BRUNA *et al.*, 2013; LECUN; BENGIO; HINTON, 2015; NIEPERT; AHMED; KUTZKOV, 2016; XIN; ZHANG; SHAO, 2018).

Na literatura, as redes neurais têm sido amplamente utilizadas para aprendizado automático de representações ou características em diversos tipos de dados. Tais representações se diferem de abordagens mais tradicionais, como as baseadas em redes complexas, que produzem atributos conhecidos como "feito a mão" (do inglês, *handcrated*) (ZHANG *et al.*, 2018; SIMON; UMA, 2020). Em relação aos métodos denominados *handcrated*, os atributos são computados por modelos matemáticos, estatísticos ou técnicas projetadas manualmente para descrever determinados aspectos do dado em questão, permitindo uma compreensão dos mecanismos relativa ao funcionamento da caracterização (NANNI; GHIDONI; BRAHNAM, 2017). De outro modo, no aprendizado de representações por redes neurais, as camadas ocultas são treinadas com grandes quantidades de dados e aprendem a extrair as principais características que descrevem o problema, variando das mais básicas até as mais específicas.

Com o propósito de se aproveitar da capacidade de aprendizado de características sofisticadas das redes neurais, nesta tese, as Redes Neurais Randomizadas (em inglês, *Randomized Neural Network* (RNN)) (SCHMIDT; KRAAIJVELD; DUIN, 1992; PAO; TAKEFUJI, 1992; PAO; PARK; SOBAJIC, 1994; HUANG; ZHU; SIEW, 2006) foram investigadas como ferramenta de caracterização das redes complexas. Em contraste com as técnicas de redes neurais profundas, a RNN se destaca pela simplicidade (uma camada oculta), algoritmo de aprendizado rápido e bom desempenho de classificação, mesmo com poucos dados de treinamento (HUANG; ZHU; SIEW, 2006; JUNIOR; BACKES, 2016). Isso torna essa rede neural uma excelente opção para o aprendizado de representações das redes complexas visando a sua caracterização, mantendo propriedades desejáveis nas técnicas desenvolvidas como simplicidade, rápido processamento e utilização de poucos dados. Nesse sentido, as atividades de pesquisa desta tese foram: (i) desenvolvimento de modelagens de imagens e vídeos em redes complexas, visando um menor número de parâmetros e melhor representação das características intrínsecas dos dados; (ii) investigação de novas técnicas de caracterização da topologia de redes complexas que aprimorem a sua descrição, melhorando o desempenho em tarefas de classificação.

1.1 Objetivos

O objetivo principal desta tese é o estudo, a proposta e análise de métodos para modelagem e caracterização de redes complexas com foco em problemas de visão computacional.

Na modelagem, foram estudadas novas formas de mapear as texturas em redes complexas direcionadas e ponderadas. Para a caracterização das redes complexas, propôs-se a investigação das seguintes metodologias: métricas clássicas, redes neurais randomizadas e autômatos de rede. Os métodos desenvolvidos para os problemas de visão computacional foram avaliados em aplicações de biologia, botânica, físico-química e medicina.

Resumidamente, os objetivos desta tese são:

- investigar e analisar os métodos da literatura de reconhecimento de padrões baseado em redes complexas e, quando possível, propor melhorias. Nesse sentido, foram propostas duas melhorias: (i) padrões binários em autômatos de rede (LLNA-BP) para caracterização de redes complexas e (ii) redes de transformada da distância para análise de formas;
- estudo e desenvolvimento de técnicas de modelagens simples e otimizadas para mapear texturas em níveis de cinza, colorida e dinâmicas em redes complexas;
- propor métodos para caracterização de redes complexas baseados em representações aprendidas com redes neurais randomizadas, combinando, assim, ambas as metodologias;
- avaliar e comparar os métodos propostos com trabalhos da literatura;
- aplicar os métodos desenvolvidos em problemas reais.

Uma visão geral do desenvolvimento desta tese é apresentada na [Figura 1](#). Os dados são modelados como redes complexas e, em seguida, é realizado a sua caracterização com uma das metodologias consideradas. Com base nesse fluxo, métodos para diversas tarefas foram desenvolvidos e, depois, aplicados em diferentes problemas reais. A cor da seta representa um fluxo específico da metodologia usada para desenvolver os métodos. Por exemplo, enquanto as setas em azul representam o método RNNDT para textura dinâmica que não utiliza rede complexa, o fluxo das setas em verde corresponde ao método LLNA-BP para classificação de redes complexas.

1.2 Contribuições

A investigação de novas formas de modelagem e caracterização de redes complexas permitiu contribuições nos âmbitos teóricos e práticos, como mostrado na [Figura 1](#) (c)-(d). Isto proporcionou uma multidisciplinaridade ao trabalho, resultando em diversas contribuições que podem ser assim sumarizadas:

- desenvolvimento de modelagens rápidas e simples que mapeiam texturas em níveis de cinza, coloridas e dinâmicas para redes complexas direcionadas e ponderadas;

Figura 1 – Visão geral do desenvolvimento desta tese. As setas representam o fluxo metodológico para o desenvolvimento dos métodos, em que cada cor representa um tipo de abordagem.

Fonte: Elaborada pelo autor.

- desenvolvimento de novas abordagens para análise de texturas que utilizam redes neurais randomizadas para caracterizar redes complexas por meio de representações aprendidas;
- proposta de dois métodos que empregam RNNs e redes complexas para caracterização de texturas dinâmicas;
- desenvolvimento de dois métodos de análise de formas baseados na combinação de redes complexas com transformada da distância e RNNs;
- caracterização e classificação de redes complexas gerais usando uma melhoria do método LLNA que emprega dicionários de padrões binários nos diagramas de padrões de evolução temporal;
- aplicações dos métodos desenvolvidos para estudar cinco problemas reais das áreas de biologia, botânica, físico-química e médica.

1.3 Organização

O presente trabalho está organizado em 8 capítulos, além desta introdução e um apêndice.

Capítulo 2 e 3: nestes capítulos são apresentados os conceitos, fundamentação teórica e trabalhos relacionados às metodologias abordadas que servem como embasamento para os métodos propostos: redes complexas, autômatos celulares e redes neurais artificiais.

Capítulo 4: este capítulo apresenta o estudo de novas formas de modelar texturas em três tipos de dados (imagens em níveis de cinza, RGB e vídeos) em redes complexas direcionadas e ponderadas.

Capítulo 5: este capítulo apresenta os métodos desenvolvidos para caracterização das redes complexas de texturas modeladas usando as abordagens descritas no capítulo anterior. Para tanto, foi proposta uma abordagem inovadora baseada em aprendizado de representações por meio do uso de RNNs. Experimentos e análises dos métodos foram realizadas com base em diversos conjuntos de dados de referência, e seus resultados foram comparados com métodos da literatura.

Capítulo 6: dois métodos propostos para análise de contorno de formas são apresentados em detalhes neste capítulo. O primeiro método incorpora a transformada da distância Euclidiana na modelagem de redes complexas, resultando em uma rede de transformada da distância. O segundo método propõe o aprendizado de representações por meio de RNNs aplicadas em mapas de características topológicas da evolução dinâmica das redes complexas de contorno. Análises e resultados obtidos pelos métodos demonstram as suas capacidades discriminativas em comparação à literatura.

Capítulo 7: neste capítulo é apresentada uma melhoria baseada em padrões binários da abordagem para caracterização de redes complexas conhecidas como autômatos de rede *Life-Like*. Para validar a abordagem, experimentos foram realizados com redes complexas sintéticas e reais.

Capítulo 8: cada seção deste capítulo apresenta uma aplicação dos métodos desenvolvidos: reconhecimento de espécies de peixes por meio do contorno do otólito, análise da plasticidade foliar de Jacarandá Caroba, detecção de câncer de próstata por meio de imagens de genossensores, detecção precoce de osteoartrite de joelho em imagens de raio-x e reconhecimento de espécies de plantas.

Capítulo 9: por fim, este capítulo apresenta as conclusões, contribuições e trabalhos futuros desta tese, bem como os artigos publicados e em redação.

Apêndice A: neste apêndice são descritas as bases de dados, os algoritmos e métricas para avaliação empregadas no trabalho.

REDES COMPLEXAS

As redes complexas ou ciência das redes (em inglês, *network science*) (BARABÁSI; PÓSFAL, 2016) emprega o formalismo da teoria dos grafos com a incorporação de métodos advindos da mecânica estatística para caracterizar sua estrutura de conexões, criar modelos e analisar a topologia e dinâmica da rede (COSTA *et al.*, 2007; CASANOVA, 2013). A teoria dos grafos tem como origem o problema das pontes de Königsberg de Euler em 1736 (EULER, 1741), cuja conjectura era verificar se era possível cruzar as sete pontes da cidade de Königsberg sem atravessar por uma mesma ponte duas vezes. Desde então, esse campo tem sido estudado em matemática e, nos últimos 60 anos, em ciência da computação. Nas últimas décadas, a teoria dos grafos atraiu a atenção de pesquisadores de diferentes áreas (como a física), que contribuíram para o campo, incorporando a análise e métodos que vieram da mecânica estatística para estudar sistemas complexos. Assim, surge o termo redes complexas, que apesar dos estudos já realizados por Erdős e Rényi (1960) em meados dos anos 1960, possui outras três investigações que causaram impacto na área (COSTA *et al.*, 2007): redes de pequeno mundo (WATTS; STROGATZ, 1998), descoberta de redes livre de escala (BARABÁSI; ALBERT, 1999) e identificação de estruturas de comunidade em redes (GIRVAN; NEWMAN, 2002).

Muitos fenômenos naturais podem ser modelados como redes complexas através da definição de um conjunto de entidades e estabelecendo um critério de relação entre estas. Alguns exemplos clássicos são a internet, composta de vários computadores e roteadores conectados, e uma rede de células, descrevendo substâncias químicas conectadas por reações químicas. Devido à capacidade de representação e característica multidisciplinar, as redes complexas têm sido usadas para estudar uma ampla variedade de sistemas em diferentes campos da ciência como, por exemplo, redes de sistemas sociais (NEWMAN; PARK, 2003), biológicos (BARABÁSI; OLTVAI, 2004), físicos (BORNHOLDT; SCHUSTER, 2006), de infraestrutura (NASIRUZZAMAN; POTA; MAHMUD, 2011), imagens (GONÇALVES *et al.*, 2016), entre outros. Esta variabilidade de aplicações e investigações em diversas áreas da ciência e sob perspectivas distintas, fazem com que a área de redes complexas avance rapidamente.

No início, as redes eram estudadas de maneira determinística, posteriormente, os matemáticos Paul Erdős e Alfred Rényi as definiram e estudaram de forma estocástica. Nesse sentido, nas últimas décadas, diversos trabalhos mostraram padrões presentes em muitas redes, que foram então entendidos como uma característica estrutural presente em alguns modelos. A partir dessas características estruturais e propriedades estatísticas, as redes podem ser classificadas em diferentes modelos, os principais são:

- **Redes aleatórias** (ERDÖS; RÉNYI, 1960): modelo mais básico gerado a partir de conexões aleatórias entre os vértices e com uma distribuição do grau dos vértices seguindo uma distribuição de probabilidade de Poisson.
- **Redes pequeno mundo** (WATTS; STROGATZ, 1998): modelo caracterizado por caminho médio pequeno entre os vértices e um número de ciclos de ordem três (i.e., três vértices conectados entre si) significativamente superior que em redes aleatórias.
- **Redes livre de escala** (BARABÁSI; ALBERT, 1999): nesse modelo, alguns poucos vértices possuem um alto número de conexões em comparação ao restante da rede, fazendo com que a distribuição do grau siga uma lei de potência.

2.1 Definição

Formalmente, uma Rede Complexa (RC) pode ser representada por um grafo $G = (V, E)$ descrito por dois conjuntos: um com N vértices $V = \{v_1, \dots, v_N\}$ e o de arestas E . Uma aresta e_{v_i, v_j} representa uma conexão entre dois vértices v_i e v_j , portanto, o conjunto $E = \{e_{v_i, v_j}, \dots\}$ é composto de todas as arestas conectando vértices de V . A rede também pode ser direcionada, ou seja, as arestas e_{v_i, v_j} possuem uma direção de v_i para v_j que mapeia informações do sistema modelado. Adicionalmente, as arestas também podem possuir pesos $w(e_{v_i, v_j})$ associados ao problema estudado (e.g., distância entre cidades, custos etc.), tornando as redes ponderadas.

Uma RC pode ser composta por diversos componentes desconectados, que também são conhecidos como subgrafos. Para determinados fins, é considerado o componente que possui o maior número de vértices conectados entre si. No âmbito computacional, as redes ou grafos podem ser representados e armazenados usando: lista de adjacência, lista de arestas e matriz de adjacência. Como exemplo, em uma matriz de adjacência A , os vértices v_i e v_j estão conectados se a posição $A(i, j)$ conter o valor 1 (ou o valor do peso da aresta), caso contrário $A(i, j) = 0$.

Na maioria dos estudos envolvendo a área, o primeiro passo é definir como modelar o problema alvo em uma rede complexa, definindo assim os conjuntos de vértices e de arestas. Uma vez que G esteja adequadamente construído, muitas medidas podem ser calculadas para quantificar sua estrutura, variando de centralidade, medidas baseadas em caminho, estrutura de comunidade e outras (COSTA *et al.*, 2007). Além disso, como em muitos casos a estrutura de uma rede real é resultado da evolução contínua das forças que a formaram (BOCCALETTI *et al.*,

2006), a sua dinâmica pode ser analisada pela caracterização da evolução estrutural em função do tempo ou de algum parâmetro de modelagem.

2.2 Medidas e Características Topológicas

Em termos de análise, diversas medidas podem ser utilizadas para estudar a estrutura e topologia das redes complexas. Essas medidas permitem a caracterização, comparação e classificação das redes que representam um determinado objeto de estudo. A seguir algumas medidas clássicas e características de redes são apresentadas.

- **Grau e grau médio:** através do grau de um vértice é possível derivar diversas medidas e análises. O grau k_{v_i} de um vértice v_i é calculado como o número de arestas incidentes a outros vértices: $k_{v_i} = \sum_{j=1}^N A(i, j)$. Por outro lado, em redes complexas ponderadas, o grau ponderado ou força s_{v_i} é computado pelo somatório dos pesos das arestas incidentes ao vértice. Já o grau médio é definido como o valor médio dos graus de todos os vértices: $\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_{v_i}$. Em redes direcionadas, também podem ser calculados: (i) grau de saída $\bar{k}_{v_i}$ dado pela soma de arestas saindo do vértice v_i e (ii) grau de entrada $\hat{k}_{v_i}$ calculado como o número de arestas de entrada em v_i . Quando as redes são direcionadas e ponderadas, outras duas medidas podem ser derivadas: (i) força de saída $\bar{s}_{v_i}$ que é a soma dos pesos das conexões de saída e (ii) força de entrada $\hat{s}_{v_i}$ dado pela soma dos pesos das arestas de entrada.
- **Distribuição do grau:** essa é uma função que indica a probabilidade de ocorrência $P(k)$ de um determinado vértice v_i ter grau k .
- **Medidas de heterogeneidade:** para medir a heterogeneidade da rede, pode-se calcular várias medidas da distribuição do grau, tais como: Entropia $k_h = \sum_i P(k_i) \log P(k_i)$, Energia $k_e = \sum_i P(k_i)^2$ e Contraste $k_c = \sum_i P(k_i) k_i^2$.

Além dessas métricas, é possível quantificar propriedades relacionadas à distância entre os vértices. Assim, o comprimento do percurso entre dois vértices é igual ao número de arestas no caminho entre eles. O caminho geodésico l_{ij} é igual ao comprimento do menor percurso entre os vértices i e j . Como medida de caracterização da rede toda, pode-se utilizar a média das distâncias geodésicas:

$$L = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N l_{ij}. \quad (2.1)$$

Uma medida mais simples é o diâmetro, que é o comprimento do maior caminho geodésico, sendo definido como: $L_{max} = \max_{i,j} \{l_{ij}\}$. Outra medida relacionada é a eficiência,

essa caracteriza globalmente a rede baseada nas distâncias geodésicas entre os seus vértices, medindo o tráfego:

$$\varepsilon = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N \frac{1}{l_{ij}} \quad (2.2)$$

Quanto às medidas de transitividade, pode ser citado o coeficiente de aglomeração, que mede o número de agrupamentos intrínsecos nas RCs. Isso ocorre quando existe uma aresta entre os vértices i e j e j está também ligado ao vértice z , logo há uma alta probabilidade de i também estar ligado a z . Essa medida é interessante para quantificar padrões de texturas em imagens, visto que tais padrões consistem na relação de um píxel com a vizinhança. Assim, o coeficiente de aglomeração C é definido como:

$$C = \frac{3 \times \text{número de triângulos na rede}}{\text{número de triplas conectadas}}, \quad (2.3)$$

em que triplas conectadas são conjuntos de três vértices conectados entre si. O fator três do numerador assegura que C está no intervalo $0 < C < 1$, sendo $C = 1$ a transitividade perfeita e $C = 0$ redes sem triângulos.

Caminhadas aleatórias também têm sido utilizadas para estudo de difusão em RCs, inclusive com aplicação em visão computacional (GONÇALVES *et al.*, 2016; RIBAS; GONÇALVES; BRUNO, 2019). Uma caminhada aleatória é definida como uma partícula saindo de cada vértice e caminhando aleatoriamente pela rede por L passos. Em cada passo, a partícula visita um vértice, e o vértice visitado no próximo passo $t + 1$ é definido por um sorteio uniforme (cada vértice possui uma mesma probabilidade) dos vértices adjacentes ao visitado no passo t , conforme a seguinte probabilidade: $p(i \rightarrow j) = \frac{1}{k_i}$. Por fim, diversas outras medidas podem ser consideradas para caracterizar as redes, tais como medidas de centralidade (e.g., *betweenness*, *closeness*, *page rank* e etc), motifs, detecção de comunidades, etc. Em Costa *et al.* (2007) uma descrição completa de medidas para caracterização de redes é realizada.

Em relação aos estudos que empregam redes complexas para reconhecimento de padrões, essencialmente, possuem a seguinte metodologia: (i) cada amostra de uma base de dados é modelada como uma rede complexa, em seguida, (ii) características topológicas de cada rede modelada compõe um vetor de características para representar a rede complexa individualmente. Assim, esse vetor de características pode ser utilizado em tarefas de classificação ou agrupamento. Essa abordagem explora a capacidade de modelagem e análise das redes complexas para integrar informações e extrair características de conjuntos de dados amplos.

Na literatura, diversos trabalhos utilizam as redes complexas como ferramenta de reconhecimento de padrões em dados de diversos problemas. Dentre os trabalhos, destaca-se a classificação de textos (MACHICAO *et al.*, 2018a; AKIMUSHKIN; AMANCIO; OLIVEIRA JR., 2017), classificação de texturas (BACKES; BRUNO, 2013; GONÇALVES *et al.*, 2016), clas-

sificação de contornos (SCABINI *et al.*, 2017), classificação de vídeos (RIBAS; GONÇALVES; BRUNO, 2019), classificação de redes metabólicas e redes sociais (MIRANDA; MACHICAO; BRUNO, 2016). As próximas seções descrevem os trabalhos e fundamentos de reconhecimento de padrões em RCs baseado em autômato (autômatos de rede) e os trabalhos de redes complexas em visão computacional, que estão diretamente relacionados com esta tese.

2.3 Autômatos de Redes

Os autômatos celulares (em inglês, *Cellular Automata* (CA)) são bem conhecidos por sua capacidade de modelar sistemas complexos e, ao mesmo tempo, produzir padrões complexos a partir de regras simples. Tais padrões emergentes podem variar de estruturas mais simples, como homogêneas, estáveis ou periódicas, a padrões caóticos, e até mesmo quando a evolução leva a estruturas complexas, como é o caso do trabalho de Wolfram (1984). Os padrões espaço-temporais formados pela dinâmica dos autômatos celulares ainda estão atraindo um grande número de pesquisadores de diversos campos, tais como, ecologistas e biólogos estudaram a formação do padrão de pigmentação dos animais, como a casca da espécie *conus textile* (WOLFRAM, 1983), bem como a formação do crescimento de tumores (POLESZCZUK; ENDERLING, 2014), propagação epidêmica (SHARMA; GUPTA, 2017), crescimento de grãos (ZINOVIEVA *et al.*, 2015), dinâmica de população de plantas (CZARAN; BARTHA, 1992), difusões de reações químicas (CHOPARD; LUTHI; DROZ, 1994) e processos de colonização (ARIJ; PARROTT, 2006). Os padrões de CAs têm sido usados mesmo em campos mais diferentes, como criptografia (MACHICAO; MARCO; BRUNO, 2012; MACHICAO; BRUNO, 2017) e, por cientistas da computação, por exemplo, como ferramentas de processamento de imagens (SILVA *et al.*, 2016; MACHICAO *et al.*, 2018; ROSIN, 2006; SILVA *et al.*, 2015).

Embora os CAs sejam tipicamente estudados em tesselações regulares, estes também podem ser explorados em estruturas irregulares, como redes complexas. Recentemente, a incorporação de dinâmicas de CAs na topologia da rede, também denominada autômatos de rede, começou a ganhar mais atenção. Os autômatos de rede são uma generalização de CAs em que uma rede (tesselação) é o ambiente onde a vida artificial é simulada. O conjunto de vértices representa as células e as conexões com outros vértices representam a vizinhança. Uma regra local governa os estados das células, a partir dos quais os padrões de evolução temporal (em inglês, *Temporal Evolution Pattern* (TEP)) podem ser visualizados pela compilação de seus estados enquanto evolui no tempo. A evolução dinâmica da RC fornece um TEP a partir do qual as propriedades intrínsecas da rede complexa podem ser extraídas para serem usadas em um contexto de reconhecimento de padrões. A abordagem de autômatos de rede baseados no Life-Like é descrita em detalhes a seguir.

2.3.1 Autômatos de Rede baseados no Life-Like

Recentemente, diversos trabalhos têm estudado os padrões de evolução temporal de um autômato em redes (ZHOU; LIPOWSKY, 2005; WU; XU; WANG, 2007; MARR; HÜTT, 2009; MIRANDA; MACHICAO; BRUNO, 2016). Formalmente, um autômato de rede $\mathcal{R}$ pode ser definido por uma tupla,

$$\mathcal{R} = \langle \mathcal{T}, \mathbf{S}, s_0, \phi \rangle.$$

A tesselação $\mathcal{T}$ de um CA é representada pela topologia de uma rede complexa, portanto, cada vértice v_i será considerado uma célula c_i que é conectada com seus k_{v_i} vizinhos. $\mathbf{S}$ é o conjunto de estados restritos a estados binários $s_i = 1$ (vivo) e $s_i = 0$ (morto). A função $s(c_i, t)$ indica o estado da célula c_i no tempo t , assim, s_0 representa a configuração inicial de todas as células e ϕ é a função de transição, isto é, a regra que rege a dinâmica do autômato na rede, definindo como os estados das células são atualizados (MIRANDA; MACHICAO; BRUNO, 2016).

O número de vizinhos de cada célula é restrito à vizinhança de cada vértice da rede. Portanto, a função de transição ϕ é dada como uma função da densidade de vizinhança $\sigma(c_i)$, definida como a proporção de vizinhos vivos. A função de transição $\phi : s(c_i, t) \rightarrow s(c_i, t + 1)$ pode ser implementada de diferentes maneiras. O método *Life-Like Network Automata* (LLNA) (MIRANDA; MACHICAO; BRUNO, 2016) foi proposto como uma forma generalizada de função de densidade que pode ser de intervalo, por exemplo, $0.22 \leq \sigma(v_i, t) < 0.33$. Para descrevê-lo, considere a função de densidade de vizinhança $\sigma(v_i, t)$ do vértice v_i como a proporção de vizinhos com estado vivo, descrita por:

$$\sigma(v_i, t) = \sigma_i = \frac{1}{k_i} \sum_{j=1}^N A(i, j) \cdot s(v_j, t), \quad (2.4)$$

em que $A(i, j)$ é a matriz de adjacência da rede e k_i é o número de vizinhos ou grau do vértice v_i .

Como o LLNA traça uma correspondência entre as funções de transição e as regras da família CA 2D Life-Like, portanto, as regras do LLNA também são descritas pela notação Bx-Sy, na forma Bx₀x₁...x₈-Sy₀y₁...y₈, em que B e S representam as condições "born" e "survive", respectivamente; e, x_x e y_y representam a combinação de dígitos que variam de 0 a 8, como B23-S3, B1357-S2468, B0345-S36, B25-S8 etc. Considerando a versão de CA 2D semi-totalístico (em inglês, *outer-totalistic*), cada célula pode ter um máximo de 9 vizinhos, incluindo ela própria (vizinhança de Moore $r = 9$). Desse modo, o LLNA também tem $2^{9+9} = 262,144$ regras de transição devido às combinações das condições de nascimento (B) e sobrevivência (S).

Formalmente, a função de transição do LLNA é definida como:

$$s(c_i, t+1) = \begin{cases} 1, & \text{Se } s(c_i, t) = 0 \text{ e } x/r \leq \sigma_i < (x+1)/r \Rightarrow \text{nasce (B)} \\ 1, & \text{Se } s(c_i, t) = 1 \text{ e } y/r \leq \sigma_i < (y+1)/r \Rightarrow \text{sobrevive (S)} \\ 0, & \text{caso contrário.} \end{cases} \quad (2.5)$$

Para fins de ilustração, é mostrado na [Figura 2](#) a dinâmica do LLNA em uma dada rede complexa contendo $N = 21$ vértices com a regra B25-S8 evoluída durante três passos de tempo. No canto superior esquerdo da [Figura 2](#) (a) é apresentado a configuração inicial s_0 da rede. Na [Figura 2](#) (b) está detalhado como as células c_i e c_j (região sombreada) são evoluídas de $t = 0$ para $t = 1$. Primeiro, considerando o estado morto da célula c_i , ou seja, $s(c_i, t) = 0$, então as duas condições de nascimento da regra dada B2 ou B5 podem ser aplicadas na [Equação 2.5](#). Como c_i tem 5 vizinhos, dos quais 3 estão vivos e 2 mortos, a sua densidade de vizinhança $\sigma(c_i, t) = \frac{3}{5} = 60\%$ satisfaz a segunda condição de nascimento $\frac{5}{9} \leq \sigma(c_i, t) < \frac{6}{9}$, portanto, no estado seguinte, ela nascerá $s(c_i, t+1) = 1$. Por outro lado, a célula c_j , com o estado $s(c_j, t) = 1$, ativa a condição de sobrevivência da regra dada S8, assim é checada a condição $\frac{8}{9} \leq \sigma(c_j, t) \leq 1$. No entanto, como $\sigma(c_j, t) = \frac{3}{4} = 75\%$, a condição de sobrevivência não é satisfeita e, portanto, no próximo estado essa célula morrerá $s(c_j, t+1) = 0$. De maneira semelhante, todos os estados do autômato também são calculados de forma síncrona. Na [Figura 2](#) (c), é possível observar o padrão de evolução temporal gerado pelo autômato, em que a evolução de cada célula é empilhada de cima para baixo. Deve-se notar que, por uma questão de visualização, os vértices da rede foram colocados da esquerda para a direita de acordo com um critério, por exemplo, rótulos dos vértices, ordenação de sua conectividade de grau, entre outros.

Descritor LLNA

Uma vez obtido um TEP da dinâmica do LLNA (veja a [Figura 2](#) (c)), cada vértice da rede pode ser analisado como uma sequência de zeros e uns, a partir dos quais várias características podem ser medidas. [Miranda, Machicao e Bruno \(2016\)](#) propuseram a combinação de três medidas para formar o vetor de características: a entropia de Shannon ($\bar{\mu}_S$), comprimento de palavra ($\bar{\mu}_W$) e a complexidade de Lempel-Ziv ($\bar{\mu}_L$). Mais especificamente, a entropia de Shannon de uma sequência binária é definida como $\mu_{S_i} = -(p_i^{(0)} \log_2 p_i^{(0)} + p_i^{(1)} \log_2 p_i^{(1)})$, onde $p_i^{(0)}$ e $p_i^{(1)}$ é a probabilidade de ter zeros e uns, respectivamente. O comprimento da palavra descreve a homogeneidade da sequência em relação ao comprimento das palavras (subsequência de uns limitadas por zeros). Finalmente, a complexidade Lempel-Ziv é uma medida baseada na contagem de todas as combinações possíveis de sequências de uns e zeros (blocos) contidas na sequência, sendo calculada como $\mu_L = g \log l / l$, em que l é o comprimento da sequência e g o número de blocos encontrados na sequência. No entanto, a abordagem proposta por [Miranda, Machicao e Bruno \(2016\)](#) considera a combinação dos histogramas de distribuição dessas métricas, o que leva à falta de um número ideal de bins.

Figura 2 – Construção do autômato de rede Life-Like usando a regra B25-S8 para uma rede com $N = 21$ evoluída durante três iterações. No topo, da esquerda para a direita, a evolução do LLNA é visualizada. Os estados são representados por branco (vivo) e preto (morto). Na parte inferior, detalha-se passo a passo a função de transição B25-S8.

(a) Regra B25-S8

(b)

$$s(c_i, 0) = 0 \Rightarrow \boxed{\text{B25}} \begin{cases} \text{if } \frac{2}{9} \leq \sigma(c_i, t) < \frac{3}{9} \\ \text{or} \\ \text{if } \frac{5}{9} \leq \sigma(c_i, t) < \frac{6}{9} \end{cases} \Rightarrow s(c_i, 1) = 1$$

$$s(c_j, 0) = 1 \Rightarrow \boxed{\text{S8}} \quad \text{if } \frac{8}{9} \leq \sigma(c_j, t) \leq 1 \Rightarrow s(c_j, 1) = 0$$

(c)

Fonte: Ribas, Machicao e Bruno (2020).

Seleção de regras LLNA

O trabalho de [Miranda, Machicao e Bruno \(2016\)](#) discute a escolha das melhores regras do LLNA para um problema específico de classificação. Os autores propuseram um procedimento de seleção de regras que consiste na busca por um conjunto de regras (soluções) que maximiza as taxas de acurácia de um problema subjacente. Esse é um processo importante, já que a mesma regra pode apresentar comportamentos diferentes para redes de classes diferentes, portanto, não é trivial prever qual seria a melhor regra para um problema específico. Nesse trabalho, também foi enfatizado que, como a seleção de regras é feita através de um procedimento de otimização, deve-se presumir que o conjunto de soluções (regras selecionadas) também pode trazer algumas regras que não atendem a expectativa ao apresentar um novo conjunto de dados. Por essa razão, cada uma das 262,144 regras da família Life-Like deve ser avaliada. Ou seja, um autômato de rede é construído para cada regra, realizando um processo de pré-classificação, isto é, vetores de características são extraídos e então é obtida uma taxa de pré-classificação. Uma vez que um conjunto de melhores regras é selecionado, o LLNA pode ser usado para as principais tarefas de classificação.

Além disso, como enfatizado por [Miranda, Machicao e Bruno \(2016\)](#), a escolha das melhores regras deve ser realizada em um subconjunto representativo da base de dados de redes complexas, separado exclusivamente para a seleção de regras, de modo a obter resultados imparciais, enquanto o restante do conjunto de dados de rede é utilizado para avaliar o desempenho dos classificadores. Dessa maneira, esse procedimento garante que uma base de dados invisível seja usado para a seleção de regras e para a validação do método, visando a imparcialidade do

esquema de validação cruzada para as etapas de treinamento e teste.

2.4 Redes Complexas em Visão Computacional

Uma vez que as redes complexas vêm sendo utilizadas como ferramentas para reconhecimento de padrões em diferentes problemas de visão computacional, nesta seção, são relatados os principais trabalhos reportados da literatura, principalmente, os relacionados às texturas e contorno de formas. Dentre os diferentes problemas de visão computacional em que o uso de redes complexas ou grafos é empregado, pode-se citar: segmentação (MORRIS; LEE; CONSTANTINIDES, 1986), análise de texturas (CHALUMEAU *et al.*, 2006; CHALUMEAU *et al.*, 2008; CANTERO *et al.*, 2018), análise de formas (BACKES; BRUNO, 2010a; SCABINI *et al.*, 2017), reconhecimento de objetos em cenas (MATAS; MARIK; KITTLER, 1995), compressão de imagens (LABELLE *et al.*, 1998), entre outros. Uma compilação completa de modelagem e medidas de redes para texturas e contorno é descrita no trabalho de Casanova (2013).

Dentre os primeiros trabalhos, o precursor em reconhecimento de padrões em imagens de texturas empregando conceitos de redes complexas foi o estudo realizado por Chalumeau *et al.* (2006), Chalumeau *et al.* (2008). Nesse estudo, uma abordagem para análise de texturas que se baseia na modelagem da imagem como uma rede complexa foi proposta para, em seguida, extrair características topológicas de grau hierárquico para caracterização. Para modelagem, cada pixel da imagem representa um vértice, que é conectado por arestas a outros vértices em uma região circular de raio r . As arestas possuem pesos relacionados a diferença de escala de cinza I entre os pixels. Os resultados alcançados por esse trabalho em conjunto de dados de texturas conceituados serviram como base para os trabalhos seguintes da área.

Posteriormente, outras abordagens variantes surgiram, tal como a proposta por Gonçalves *et al.* (2012). Esse trabalho utiliza uma técnica de limiarização das arestas para estudar a dinâmica da RC ao longo de diferentes transformações e, assim, realizar uma análise multi-escala da imagem de textura. Para caracterizar essa dinâmica de rede, é utilizado o transiente e atrator das trajetórias produzidas pela caminhada determinística parcialmente auto-repulsiva ou caminhadas do turista sobre a RC. Por outro lado, Backes, Casanova e Bruno (2013) propuseram uma modelagem em que os pesos das arestas sejam definidos por uma combinação entre a distância Euclidiana dos pixels e a diferença de escala de cinza, incorporando novas informações na RC. Em seguida, as redes modeladas e transformadas por limiares são caracterizadas pelas medidas de grau médio, entropia, energia e contraste da distribuição dos graus dos vértices. Nesse trabalho, a técnica de modelagem e as medidas usadas demonstraram através de experimentos serem robustas para a análise de imagens rotacionadas e com ruído.

De modo a superar limitações de parametrização na etapa de modelagem, Backes, Martinez e Bruno (2011) propuseram uma modelagem inovadora baseada na utilização das trajetórias da caminhada determinística do turista na imagem. Dessa forma, esse trabalho contribuiu para a

redução no número de parâmetros necessários para modelagem, embora as caminhadas determinísticas também aumente o custo computacional para construção da RC. Resumidamente, o funcionamento dessa técnica pode ser descrita como: (i) primeiramente, o conjunto de arestas E da RC é vazio; (ii) cada pixel, representando um vértice, é um ponto de início para uma caminhada do turista baseada em uma regra determinística e uma memória; (iii) uma aresta é adicionada a cada movimento do turista na imagem; (iv) o vetor de características é formado pelo grau médio e entropia do grau conjunto das redes construídas. Assim, as conexões entre os vértices são construídas conforme os movimentos do turista nos pixels das imagens. Nesse sentido, outros trabalhos também combinam RC e caminhadas determinísticas com sucesso para análise de imagens (COUTO *et al.*, 2015; GONÇALVES *et al.*, 2012).

No trabalho proposto por Gonçalves *et al.* (2016) duas contribuições são introduzidas: a modelagem de imagens de texturas em redes complexas direcionadas e o uso de difusão para caracterização das redes modeladas. Na modelagem, as arestas com peso negativo são removidas, assim, a direção da aresta é determinada pela direção do gradiente: se $I(v_i) > I(v_j)$, então a aresta sai de v_i para v_j . Para caracterização da RC, uma série de transformações são aplicadas para remover arestas e em cada rede transformada a difusão é aplicada para estimar a atividade dos vértices. Basicamente, a difusão é realizada da seguinte forma: em cada vértice é iniciado uma caminhada aleatória e a frequência relativa de visitas do vértice é a sua atividade. O vetor de características é composto por um histograma de atividade que calcula a atividade de todos os vértices com um dado grau de entrada. A principal desvantagem do método é o custo computacional para realizar as caminhadas aleatórias em um conjunto de RCs geradas por diferentes raios e transformações.

Por outro lado, Junior, Backes e Cortez (2013) modelam a imagem em uma rede não direcionada com os pesos das arestas determinados por: $|I(v_i) - I(v_j)| + \frac{I(v_i) - I(v_j)}{2}$. Medidas de distância geodésica entre todos os vértices usando diferentes escalas e orientações são empregadas para caracterizar a textura. Experimentos em bases de texturas mostraram que o método é robusto a rotação e mudança de perspectiva. Ainda em texturas em níveis de cinza, uma abordagem baseada em redes complexas direcionadas e importância dos vértices foi desenvolvida por Cantero *et al.* (2018). A principal contribuição desse trabalho foi calcular a métrica de *pagerank* para RC de texturas visando correlacionar a importância do vértice com o seu grau, tornando essa medida um poderoso extrator de características de texturas, especialmente para reconhecimento de doenças em folhas de soja.

Em texturas coloridas, Scabini *et al.* (2019a) desenvolveram uma abordagem de rede complexa multi-camada para caracterizar imagens de texturas coloridas. Essa técnica modela a imagem colorida da textura interligando os diferentes canais RGB com arestas e usa medidas clássicas da topologia dessa RC para construir o vetor de características. Em seguida, uma segunda abordagem também foi proposta por Scabini, Ribas e Bruno (2020) para modelar texturas coloridas em imagens RGB como redes complexas direcionadas espaço-espectral. Tal

modelagem define para as arestas uma direção baseada no gradiente e um peso dado por uma função que combina as informações de distância e cor. Além disso, a topologia inclui conexões entre diferentes canais para analisar as características espectrais, enquanto conexões no mesmo canal mapeiam propriedades espaciais dos canais individuais (SCABINI; RIBAS; BRUNO, 2020). Para caracterizar as RCs, seis medidas estatísticas da distribuição de grau de saída, força de saída e força de entrada de toda a RC, das conexões espaciais e espectrais são calculadas para compor o vetor de características. Os resultados alcançados em bases de dados conceituadas de texturas coloridas demonstraram que esse método é competitivo quando comparado aos demais da literatura, incluindo baseados em aprendizado profundo.

Em relação à análise de contorno de forma, Backes, Casanova e Bruno (2009a) introduziram uma nova metodologia para análise de formas usando redes complexas. Dado uma imagem de contorno, essa é modelada em uma rede regular mapeando cada ponto do contorno em um vértice e conectando todos os vértices entre si. Para cada aresta, é calculado um peso baseado na distância normalizada entre os pontos do contorno. Uma vez construída a rede, uma série de transformações são aplicadas em suas arestas, fazendo com que sua topologia se aproxime do modelo de redes pequeno mundo em vários estágios. Em seguida, medidas de grau e correlação de grau ao longo dos estágios de transformações são usadas para compor o vetor de características. Os resultados obtidos em reconhecimento de formas superaram métodos tradicionais da literatura, com destaque para diversas características desejáveis demonstradas pelo método, tais como robustez à degradação, tolerância contra ruído e invariância à rotação e escala. Outro trabalho derivado desse foi proposto por Backes, Martinez e Bruno (2010). Esse trabalho combina redes complexas e fractal multi-escala para extrair características da forma. Scabini *et al.* (2017) propuseram um método para reconhecimento de formas baseado em descritores angulares de redes complexas. Basicamente, ao invés de utilizar medidas de grau, os autores calcularam os ângulos das arestas da rede para compor o vetor de características. Os resultados mostram uma melhoria na caracterização em relação aos trabalhos anteriores, indicando a necessidade de investigar formas mais robustas de caracterizar a rede complexa.

Recentemente, Gonçalves, Machado e Bruno (2015) realizaram um estudo precursor de redes complexas para reconhecimento de padrões em texturas dinâmicas com resultados promissores. Nesse trabalho, o vídeo da textura é modelado em uma rede regular, conectando os vértices em um raio volumétrico e atribuindo para cada aresta um peso dado pela diferença entre as intensidades de níveis de cinza dos pixels. Em um segundo passo, a dinâmica de evolução da RC para uma série de limiares é analisada considerando o grau espacial (conexões no mesmo quadro) e temporal (conexões entre quadros diferentes). Apesar de simples, essas medidas descrevem adequadamente diferentes características de aparência e de movimento na textura. Os resultados demonstram o potencial do método em análise de texturas dinâmicas sintéticas e reais. Em um segundo trabalho desenvolvido por Ribas, Gonçalves e Bruno (2019), as texturas dinâmicas foram estudadas usando uma modelagem de redes complexas direcionadas e difusão para caracterização. Outro trabalho usando redes complexas geradas por caminhadas

determinísticas do turista também foi proposto para analisar e reconhecer texturas dinâmicas (RIBAS; BRUNO, 2020). Apesar dos resultados promissores, esses trabalhos têm como limitação o número de parâmetros de modelagem ou a simplicidade das medidas de caracterização da topologia da RC.

Usualmente, as técnicas de reconhecimento de padrões baseadas em RCs empregam a seguinte metodologia: modelagem da imagem/vídeo em uma rede complexa para, posteriormente, utilizar seus aspectos topológicos para caracterização. A maioria dos métodos da literatura tem como principal desvantagem o número de parâmetros relacionados à modelagem que precisam ser calibrados. Nesse sentido, alguns trabalhos (RIBEIRO *et al.*, 2015; SCABINI; GONÇALVES; JR, 2015; SCABINI *et al.*, 2019a) propuseram maneiras de selecionar automaticamente os parâmetros para agilizar o processo de calibração dos métodos. Assim, a definição do modelo para construção da rede complexa é uma parte importante para a obtenção de um descritor competitivo. Na última década, diversos modelos para mapeamento de imagem em RC vêm sendo estudados pela comunidade de visão computacional. Essencialmente, a ideia por trás da etapa de modelagem é: dada uma imagem, mapeá-las em um conjunto de vértices V e arestas E , que possuem pesos definidos por uma função de ponderação. Para tanto, é preciso determinar os dois conjuntos: vértices V e arestas E . Um estudo de uma ampla variedade de possíveis definições desses conjuntos é realizado por Casanova (2013). A seguir, é listada uma compilação com as principais definições de ambos os conjuntos e da função de peso:

- **Vértices:** cada vértice $v \in V$ representa um subconjunto de pixels S_i que pertence ao conjunto universo C . Tal subconjunto S_i pode representar: um único pixel, um objeto, um conjunto de pixels adjacentes (e.g., uma região 4-conectada ou 8-conectada), um conjunto de pixels que não são adjacentes aos outros (e.g., conjunto de pixels com a mesma cor).
- **Arestas:** essa definição concerne à construção das relações de adjacência entre os vértices. Apesar dessa definição ser diretamente relacionada à função de ponderação, algumas construções são consideradas, tal como: *lattice 4-conectada* (pixels são mapeados em vértices, e arestas são adicionadas em vizinhanças de 4 pixels na imagem); *lattice 8-conectada* (arestas são atribuídas aos vértices se eles pertencem a uma vizinhança de 8 pixels na imagem); *lattice completa* (todos os vértices são conectados entre si); *lattice conectada por raio* (conecta os vértices se os pixels representados por eles estão dentro de um dado raio r); *lattice k-vizinhos* (cada vértice é ligado aos k -vizinhos mais próximos baseado em uma regra de distância); *adjacência por regiões* (conecta vértices que representam regiões vizinhas); *adjacência por cor* (um par de vértices é conectado quando as suas intensidades de cor são adjacentes na imagem).
- **Função de peso:** a definição da função de peso das arestas está diretamente associada a regra de adjacência. Algumas definições da função de peso $\delta(e_{v_i, v_j})$ são:

- distância Euclidiana entre dois vértices/pixels v_i e v_j :

$$w(e_{v_i, v_j}) = \delta(e_{v_i, v_j}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

- diferença de intensidade de cor normalizada: $w(e_{v_i, v_j}) = \delta(e_{v_i, v_j}) = \frac{I_i - I_j}{L}$

- combinação da distância e diferença de escala de cinza normalizada entre um par de vértices v_i e v_j :

$$w(e_{v_i, v_j}) = \delta(e_{v_i, v_j}) = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{2r} + \frac{I_i - I_j}{2L}$$

- combinação da distância Euclidiana e diferença de intensidade de cinza normalizada entre um par de vértices/pixels: $w(e_{v_i, v_j}) = \delta(e_{v_i, v_j}) = \frac{(x_i - x_j)^2 + (y_i - y_j)^2 + r^2 \frac{|I_i - I_j|}{L}}{r^2 + r^2}$, em que I_i e I_j denotam a intensidade de cor dos pixels i e j ; x e y são os índices dos pixels; L é o nível máximo de cor da imagem e r é o raio de conexão usado.

Dada uma imagem modelada baseado nas definições descritas acima, medidas de sua topologia são extraídas de diferentes maneiras. A [Figura 3](#) mostra exemplos de textura estática, textura dinâmica e contorno modelados como uma rede complexa. Na figura, os pixel são representados por vértices e as arestas conectam-os baseado em um raio de distância. A [Figura 3\(a\)](#) ilustra uma textura em níveis de cinza em que regiões homogêneas da imagem formam pequenos conjuntos de vértices conectados. Isso ocorre porque nesta modelagem limiares são aplicados para manter apenas arestas conectando pixels similares. Já na [Figura 3\(c\)](#), que exemplifica a modelagem de textura dinâmica em redes complexas, a cor dos vértices ilustra os quadros do vídeo. Nesse caso, as conexões são no mesmo quadro e entre diferentes quadros.

Figura 3 – Ilustração de modelagens em redes complexas. (a) Textura estática, (b) contorno de forma, (c) textura dinâmica. Os pixels são representados por vértices e as arestas conectam os vértices de acordo com um raio de distância e similaridade de cor.

Fonte: Elaborada pelo autor.

REDES NEURAIS ARTIFICIAIS

Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados nos mecanismos por trás do cérebro de mamíferos, que é altamente complexo, não-linear e paralelo, além de ser composto por bilhões de neurônios, que são organizados para realizar determinadas tarefas (e.g., reconhecimento de padrões) muito mais rapidamente que os computadores convencionais (HAYKIN, 2007; GAMA *et al.*, 2011). Similarmente, as RNAs são formadas por centenas ou até mesmo milhares de unidades de processamento, conhecidas também como neurônios, e adquirem conhecimento baseado em experiência, ou seja, por meio de dados rotulados previamente por especialistas. Esses neurônios artificiais, cujo funcionamento é bem simples, são conectados por sinais de entrada que possuem pesos. Os neurônios realizam operações locais sobre seus dados de entrada recebidos pelas suas conexões e as iterações entre os neurônios produzem o comportamento inteligente de uma rede neural artificial (GAMA *et al.*, 2011).

O primeiro modelo matemático de neurônio foi desenvolvido por McCulloch e Pitts (1943), conseguindo resultados encorajadores, mas com os pesos e limiares determinados analiticamente. Posteriormente, um novo avanço na área propôs o modelo perceptron, introduzido por Rosenblatt (1958). Tal modelo teve como principal contribuição em relação ao modelo McCullough-Pitts a incorporação de um algoritmo para treinamento dos pesos. Basicamente, o modelo perceptron pode ser descrito da seguinte forma (GAMA *et al.*, 2011; HAYKIN, 2007; SILVA; SPATTI; FLAUZINO, 2010): (i) sinais de entrada são apresentados – (ii) cada sinal x_j na entrada j é multiplicado por um peso sináptico w_j – (iii) um potencial de ativação ($u = \sum_j^n w_j \cdot x_j - \theta$) é determinado por um somatório ponderado dos valores de entrada, subtraindo-se o bias θ que é o limiar de ativação – (iv) uma função de ativação $g(\cdot)$ apropriada é aplicada para limitar a saída do neurônio, produzindo uma saída $y = g(u)$. Os passos descritos acima estão ilustrados na Figura 4.

Após a popularidade inicial, falhas encontradas no modelo perceptron (e.g., sua incapacidade de lidar com problemas não-linearmente separáveis) causaram um declínio no interesse das redes neurais na década de 1970 (MINSKY; PAPERT, 2017). Mais tarde, trabalhos rea-

Figura 4 – Esquema do modelo de neurônio perceptron proposto por Rosenblatt (1958).

Fonte: Elaborada pelo autor.

lizados na década de 1980 atraíram novamente a atenção de cientistas da computação para o estudo de RNAs. Dentre os trabalhos, as propostas de novas arquiteturas com maior capacidade de representação foram determinantes, tais como, o uso de redes simétricas para otimização proposto por Hopfield (1982) e a introdução do algoritmo de aprendizado *backpropagation* por Rumelhart, Hinton e Williams (1986). No entanto, a pesquisa na área de RNAs sempre foi limitada por questões de alto custo computacional, falta de justificativas e explicabilidade para uma rede alcançar um determinado resultado, e a dificuldade na definição de uma arquitetura ideal (i.e., definir o número de camadas intermediárias e neurônios) que tenha uma boa relação entre acurácia de classificação e custo computacional (GAMA *et al.*, 2011). Mais recentemente, com avanços científicos e o surgimento de dispositivos computacionais mais poderosos, as RNAs voltaram a atrair a atenção de pesquisadores da área de inteligência artificial.

As seções seguintes apresentam alguns modelos de RNAs e seus fundamentos, que são necessários para compreender o funcionamento da etapa de aprendizado de representações presente nos métodos desenvolvidos. Inicialmente, a rede neural randomizada é descrita, esta tem como principal característica a simplicidade conceitual e algoritmo de aprendizagem rápido. Esse modelo de RNA foi adotado neste trabalho para aprender as representações das redes complexas. Adicionalmente, também são apresentados alguns conceitos de redes neurais profundas, também conhecidas pelo termo em inglês *deep learning*, para compreender a revolução que esse tópico causou na área de inteligência artificial. Alguns modelos de redes neurais profundas também foram utilizadas nos experimentos do trabalho.

3.1 Redes Neurais Randomizadas

Os primeiros estudos de redes neurais randomizadas (em inglês, *Randomized Neural Network* (RNN)), também conhecidas na literatura como *Random Vector Functional Link Neural Network*, foram realizados na década de 1990 (SCHMIDT; KRAAIJVELD; DUIN, 1992; PAO; TAKEFUJI, 1992; PAO; PARK; SOBAJIC, 1994). Posteriormente, um novo trabalho publicado

por Huang, Zhu e Siew (2006) renomeando a área como *Extreme Learning Machine*, ajudaram a atrair novamente a atenção de pesquisadores e consolidar essa arquitetura de RNA. Devido à sua simplicidade, o algoritmo de aprendizado extremamente rápido e os bons resultados alcançados, atualmente, as RNNs têm atraído a atenção de diversos pesquisadores para tarefas de reconhecimento de padrões (BACCIU *et al.*, 2018; DUDEK, 2019; PRATAMA *et al.*, 2017; ZHANG *et al.*, 2019; KAYA; TEKIN, 2013; MALAR *et al.*, 2012). Particularmente, em visão computacional, diversos trabalhos empregam as RNNs em tarefas, tais como, de classificação de microestruturas de liga de titânio (JUNIOR; BACKES; BRUNO, 2018), detecção automatizada de câncer de mama em mamografias (MUDULI; DASH; MAJHI, 2020), segmentação de imagens (ANDRUSHIA; THANGARAJAN, 2020), análise de vídeos (GUANGLI *et al.*, 2019), classificação de contorno de formas (JUNIOR; BACKES; BRUNO, 2018) e análise de texturas dinâmicas (Sá Junior; RIBAS; BRUNO, 2019). Tais trabalhos têm obtido sucesso com a proposta de abordagens que utilizam os pesos treinados da camada de saída como assinaturas, demonstrando que esta é uma linha de pesquisa com bastante potencial.

A RNN na sua forma básica tem uma única camada oculta *feedforward*, além de uma camada de entrada e uma camada de saída. Nessa arquitetura, os pesos dos neurônios ocultos são determinados aleatoriamente baseado em uma distribuição de probabilidade com o propósito de projetar não-linearmente os dados de entrada em outro espaço dimensional em que é mais provável que os vetores de atributos sejam linearmente separáveis (RIBAS *et al.*, 2020), conforme estabelecido no teorema de Cover (COVER, 1965). Por sua vez, na camada de saída, os pesos são calculados baseado no método dos mínimos quadrados visando separar linearmente esses vetores projetados. Essa solução torna o processo de aprendizado mais rápido e, portanto, permite que essas RNAs lidem com problemas que exigem mais velocidade de processamento e dispositivos computacionais mais simples, como em aplicações que envolvem o reconhecimento em tempo real.

Para descrever matematicamente o funcionamento da RNN, considere para o treinamento: $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N]$ como uma matriz de N vetores de entrada com p atributos cada e $Y = [\vec{y}_1, \vec{y}_2, \dots, \vec{y}_N]$ como uma matriz com os seus vetores de rótulo correspondentes. Uma vez que o conjunto de dados de treinamento é estabelecido, o próximo passo é obter a matriz de pesos da camada oculta H de dimensão $Q \times (p + 1)$, em que Q é o número de neurônios ocultos. Os valores dessa matriz podem ser gerados usando uma distribuição uniforme ou Gaussiana. A Figura 5 ilustra a topologia da RNN.

Uma constante -1 é adicionada em cada vetor de entrada $\vec{x}_i$ como o primeiro atributo para se conectar aos pesos do bias dos neurônios ocultos. Em seguida, a saída de cada neurônio oculto é calculada usando a função de ativação $\phi(HX)$, que é geralmente uma tangente hiperbólica ou sigmoide. Essa saída de vetores projetados da função $\phi(\cdot)$ é usada para compor uma matriz de vetores $Z = [\vec{z}_1, \vec{z}_2, \dots, \vec{z}_N]$ que será usada como entrada na camada de saída. Novamente, a constante -1 é adicionada como o primeiro atributo de cada vetor $\vec{z}_i$ para se conectar ao bias dos

Figura 5 – Esquema de uma rede neural randomizada com uma camada oculta.

Fonte: Elaborada pelo autor.

neurônios de saída.

Finalmente, os pesos dos neurônios de saída são organizados como uma matriz para a tarefa de classificação ou um vetor f no caso de regressão (i.e., apenas um neurônio de saída), na qual cada linha representa um neurônio de saída. Esses pesos são calculados visando satisfazer a equação $Y = fZ$. Para esse propósito, pode-se usar a pseudo-inversa de Moore-Penrose:

$$f = YZ^T(ZZ^T)^{-1}, \quad (3.1)$$

em que Z^T representa a matriz inversa. Em muitos casos, é comum que a matriz ZZ^T se torne singular ou quase singular e, portanto, não possa ser invertida, produzindo resultados imprecisos. Para evitar esse problema, é possível empregar a regularização de Tikhonov (CALVETTI *et al.*, 2000; TIKHONOV, 1963), resultando na seguinte equação,

$$f = YZ^T(ZZ^T + \lambda I)^{-1}, \quad (3.2)$$

onde $\lambda > 0$ e I é a matriz identidade. Note que diferentemente de outras arquiteturas de RNA que utilizam algoritmos de treinamento iterativos tal como de retropropagação, o aprendizado nesta rede neural é realizado pelo cálculo de uma única equação, tornando esse processo extremamente rápido. Por fim, usando as definições descritas acima, a RNN poderá ser treinada e utilizada para predição em um problema de classificação ou regressão.

3.1.1 Complexidade Computacional

Para avaliar a complexidade computacional da RNN, considere um conjunto de N vetores de atributos de entrada. O maior custo computacional desta rede neural é resolver a equação $\vec{f} = YZ^T(ZZ^T)^{-1}$, assim é possível analisá-la em alguns passos chaves. Primeiramente, para calcular a saída da camada oculta $Z = \phi(HX)$ e transpor Z é necessário cerca de $T_1 = Q(p+1)N$ e $T_2 = (Q+1)N$ operações, respectivamente. Em seguida, para multiplicar YZ^T , é preciso $T_3 = N(Q+1)$ operações, enquanto no cálculo de ZZ^T e da sua inversa, são necessários $T_4 = (Q+1)^2N$ e $T_5 = (Q+1)^3$ operações, respectivamente (existem métodos mais eficientes para calcular T_5). Por fim, a multiplicação dos dois termos YZ^T e $(ZZ^T)^{-1}$ requer cerca de $T_6 = (Q+1)^2$ operações. Assim, é possível considerar a complexidade total da RNN como $T = O(N)$, uma vez Q e p geralmente são bem menores que N e não dependem do número de vetores de entrada, por exemplo, em alguns casos dos experimentos, $N = 10000$, $p = 8$ e $Q = 39$. É importante ressaltar que as operações de regularização de Tikhonov e geração dos pesos aleatórios H são suprimidas para simplificar a demonstração, visto que não alteram a complexidade de tempo da RNN.

3.2 Redes Neurais Profundas

Redes neurais profundas, também conhecidas pelo termo aprendizado profundo (em inglês, *Deep Learning* (DL)), é um tópico altamente pesquisado na área de inteligência artificial que, geralmente, dispensa o uso de extratores de características. Esse tipo de aprendizado envolve todas as RNAs com pelo menos uma camada oculta, embora tenha sido demonstrado empiricamente que arquiteturas com um número maior de camadas alcançam melhor desempenho (SZEGEDY *et al.*, 2015). Ao longo dos últimos anos, importantes áreas de pesquisa tem-se beneficiado das técnicas de aprendizado profundo, tais como, visão computacional, processamento de linguagem natural e bioinformática. Em particular, na área de visão computacional, as redes neurais convolucionais (do inglês, *Convolutional Neural Network* (CNN)) têm sido amplamente utilizadas na última década.

Os primeiros modelos que serviram de inspiração para as CNNs atuais foram introduzidos durante a década de 1980, com destaque para a proposta da rede neural *Neocognitron* (FUKUSHIMA; MIYAKE, 1982) e, depois, com o trabalho proposto por Rumelhart *et al.* (1986) que apresentou o algoritmo *backpropagation*. Posteriormente, na década de 1990, um aumento no número de trabalhos resultou em melhorias que contribuíram para o surgimento das redes *Long Short-Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) e LENET-5 (LECUN *et al.*, 1989; LECUN *et al.*, 1998) para reconhecimento de dígitos.

Apesar dos resultados promissores, acreditava-se que o custo computacional para treinamento das RNAs era elevado considerando os dispositivos computacionais disponíveis na época. Após um período de pouco interesse, as RNAs só voltaram a atrair atenção da comunidade

científica após a publicação do modelo *Deep Belief Network* (DBN) (HINTON; OSINDERO; TEH, 2006) que poderia ser treinado eficientemente e, principalmente, pelo desenvolvimento de unidades de processamento gráfico (do inglês, *Graphical Processing Units* (GPU)) poderosas que impulsionaram avanços na área. Em especial, o poder das GPUs foi demonstrado a partir dos ótimos resultados alcançados pela arquitetura AlexNet proposta por Krizhevsky, Sutskever e Hinton (2012), que possui 650.000 neurônios divididos em 5 camadas convolucionais e 3 totalmente conectadas. Esse trabalho causou uma revolução na área de RNAs e inteligência artificial, contribuindo para o surgimento de arquiteturas cada vez maiores, complexas e poderosas.

Embora utilizadas em várias áreas, as técnicas de aprendizado profundo ganharam destaque em tarefas de visão computacional após seus resultados bem sucedidos nos concursos *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) (RUSSAKOVSKY *et al.*, 2015) e Pascal VOC (EVERINGHAM *et al.*, 2015). Assim, desde o trabalho introduzindo a arquitetura AlexNet em 2012, este tipo de abordagem se tornou dominante, produzindo um grande número de artigos sobre o tópico (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; GIRSHICK *et al.*, 2014; SZEGEDY *et al.*, 2015; HE *et al.*, 2016; HUANG; LIU; WEINBERGER, 2016) e aplicações em uma enorme variedade de problemas, tais como, em reconhecimento de objetos (HE *et al.*, 2016), classificação de texturas (SCABINI *et al.*, 2019b), reconhecimento de faces (MASI *et al.*, 2018) etc. Tipicamente, os maiores avanços nas arquiteturas são realizados na parte inicial, que é composta, nos primeiros modelos, por um conjunto pequeno de camadas convolucionais. Por outro lado, os modelos mais recentes aumentaram consideravelmente o número de camadas e desenvolveram novas estruturas convolucionais, como módulos *inception* na rede InceptionV3 (SZEGEDY *et al.*, 2016) e conexões residuais na arquitetura ResNet (HE *et al.*, 2016). A seguir são apresentadas algumas definições de CNNs, seus principais modelos, as técnicas de transferência de aprendizado e, por último, trabalhos que envolvem a combinação de redes neurais profundas e redes complexas (ou grafos).

3.2.1 Conceitos e Principais Modelos de CNN

As CNNs são usadas principalmente para analisar dados de alta dimensão organizados matricialmente, tais como, imagens e vídeos. Diferentemente das RNAs mais tradicionais, as CNNs profundas são caracterizadas pelas grandes arquiteturas compostas por uma sequência de muitas camadas combinadas com a aplicação de operações, tais como, *pooling*, *dropout* etc. Tais camadas aprendem as características presentes nas imagens de forma hierárquica em cada estágio. Assim, enquanto as primeiras camadas extraem informações genéricas como pontos e cantos, as camadas subsequentes combinam essas informações para extrair padrões como bordas e formas, e as últimas camadas identificam as informações específicas do problema de interesse. A seguir são listados os principais componentes da arquitetura das CNNs, bem como suas funções.

- **Camada convolucional:** essa é a camada mais importante da CNN. Basicamente, ela é

composta por um banco de filtros que são treinados. Um filtro é uma matriz (por exemplo, $3 \times 3 \times 3$) contendo valores reais que podem ser entendidos como os pesos da rede neural. Cada filtro está conectado com apenas uma região da camada anterior. Assim, esses filtros são convoluídos com os mapas de características das camadas anteriores. Os valores dos filtros são alterados à medida que o processo de treinamento é realizado, indicando regiões relevantes das imagens para a extração de características. As saídas dos filtros passam por uma função de ativação linear ou não-linear (e.g., sigmóide, softmax etc), resultando no mapa de características de saída (ALOM *et al.*, 2018).

- **Camada de pooling:** geralmente, após uma camada convolucional existe uma camada de *pooling*, também conhecida como *subsampling*. Essa técnica reduz o número de parâmetros da CNN, o custo computacional e também contribui para prevenir o *overfitting* (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; WU; GU, 2015; TRIPATHI *et al.*, 2017). Nessa camada, é aplicado uma técnica de redução do tamanho da dimensão dos mapas de saída resultantes da etapa anterior de convolução. Por exemplo, se uma máscara de amostragem 2×2 for usada, a dimensão de cada mapa de saída será reduzida pela metade. Para isso, duas operações podem ser usadas nessa camada: *average pooling* ou *max-pooling*. No *average pooling*, o valor médio presente na máscara é usada para gerar o mapa de características resultante, enquanto no *max-pooling* é utilizado o valor máximo.
- **Camada de classificação:** essa é uma camada completamente conectada com a camada anterior, diferentemente da camada convolucional na qual os filtros são conectados apenas em uma região. O funcionamento dessa camada é similar a uma rede neural tradicional, assim, a probabilidade de cada classe é determinada baseada nas características extraídas a partir das camadas anteriores. Em geral, não há uma regra sobre o número de camadas totalmente conectadas na rede, no entanto, esse número é baixo devido ao seu alto custo computacional (por exemplo, de duas a quatro camadas foram usadas nas arquiteturas LeNet, AlexNet e VGG) (ALOM *et al.*, 2018).

A arquitetura de uma CNN é definida pela organização dos componentes descritos acima, bem como outros mais específicos. Basicamente, as arquiteturas consistem em uma camada de entrada, seguida de uma pilha de camadas convolucionais e de *pooling*, e finalizada com uma ou mais camadas totalmente conectadas. A Figura 6 ilustra a organização das oito camadas da arquitetura AlexNet, que é dividida em: cinco camadas convolucionais e três camadas totalmente conectadas.

Os avanços mais proeminentes na área de CNNs são produzidos por grandes arquiteturas que consistem em combinações inteligentes de camadas mais a aplicação de algumas operações úteis entre elas, como *Dropout* (SRIVASTAVA *et al.*, 2014), *Rectified Linear Units* (ReLU) (NAIR; HINTON, 2010), *Residual Blocks* (HE *et al.*, 2016) e outras operações que visam reduzir o tempo de treinamento e evitar o *overfitting*. Alguns exemplos de arquiteturas mais

avanzadas com algumas diferenças topológicas que foram propostas nos últimos anos são: VGG (SIMONYAN; ZISSERMAN, 2014), GoogLeNet (SZEGEDY *et al.*, 2015), ResNet (HE *et al.*, 2016) e DenseNet (HUANG; LIU; WEINBERGER, 2016), PNASNet5 (LIU *et al.*, 2018) e MobileNet (HOWARD *et al.*, 2017). A seguir os modelos de CNN mais recentes e populares são apresentados.

- **VGG:** esse modelo de CNN desenvolvido pelo *Visual Geometry Group* ficou em segundo lugar no concurso ILSVRC (SIMONYAN; ZISSERMAN, 2014). A sua principal contribuição foi demonstrar que a profundidade da rede é uma questão crítica para alcançar bons resultados (ALOM *et al.*, 2018). Entre as versões propostas, as mais importantes e utilizadas são a VGG-16 e VGG-19 que possuem 16 e 19 camadas, respectivamente. Essas versões são divididas em cinco módulos que possuem uma camada de *max-pooling* no final de cada módulo. Tais módulos são seguidos de várias camadas totalmente conectadas usando a função de ativação ReLU e uma camada final softmax para classificação. Assim, a diferença entre essas duas versões é o número de camadas convolucionais presentes nos módulos. À medida que o tamanho do mapa de características é reduzido pelo *max-pooling*, o número de filtros (que são todos de tamanho 3×3) aumenta nas camadas convolucionais. Em termos de resultados, a VGG-19 é ligeiramente melhor do que a VGG-16 para classificação de imagens na base de dados ImageNet. Embora obtenha bons resultados, este modelo tem como principal desvantagem o elevado número de parâmetros que precisam ser treinados. A maioria desses parâmetros estão concentrados nas três últimas camadas totalmente conectadas, fazendo com que a maior parte do tempo de treinamento seja gasto na otimização dos pesos dessas últimas camadas.
- **GoogLeNet:** esse modelo foi proposto por Szegedy *et al.* (2015) após ganhar o ILSVRC 2014. As contribuições que tornaram-o popular foram: (i) redução da complexidade computacional através da adoção de uma única camada totalmente conectada e (ii) a incorporação de módulos *inception* que tem campos receptíveis variáveis (ALOM *et al.*, 2018). Esses módulos *inception* podem ser interpretados como sub-redes que executam operações diferentes sobre um mesmo mapa de características. Ao final, a saída resultante de todas sub-redes são conectadas. No total, a rede GoogLeNet é composta de 22 camadas, que era um número superior aos modelos anteriores da época. No entanto, uma vez que esse modelo tinha apenas uma camada conectada, que concentra a maior parte dos parâmetros, o número de parâmetros utilizado foi bem inferior à AlexNet e VGG. Existem quatro versões do modelo GoogLeNet: *Inception-v1* (SZEGEDY *et al.*, 2015), *Inception-v2* (SZEGEDY *et al.*, 2016), *Inception-v3* (SZEGEDY *et al.*, 2016) e *Inception-v4* (SZEGEDY *et al.*, 2017).
- **Microsoft ResNet:** o modelo de CNN *Residual Network* (ResNet) (HE *et al.*, 2016) foi o ganhador do ILSVRC 2015. Essa arquitetura foi desenvolvida inspirada na simplicidade

do VGG, mas com maior profundidade. Foram desenvolvidos modelos de ResNet com diferentes números de camadas: 34, 50, 101, 152 e até 1202. A ResNet50 é a mais popular e possui 49 camadas de convolução e 1 totalmente conectada (ALOM *et al.*, 2018). O modelo consiste em vários *residual blocks* que permitiram o uso de um grande número de camadas ocultas. Os *residual blocks* resolvem parcialmente o problema de *Vanishing Gradient* sofrido por arquiteturas profundas. As outras questões relacionadas ao modelo ResNet são parecidas aos demais modelos como, por exemplo, o uso da ReLU após cada camada convolucional.

Figura 6 – Organização das camadas da arquitetura AlexNet.

Fonte: Adaptada de Krizhevsky, Sutskever e Hinton (2012).

3.2.2 Transferência de Aprendizado

Embora o problema de tempo computacional tenha sido em parte solucionado com o surgimento de GPUs, uma limitação ainda persistente nas CNNs é o grande número de amostras necessárias no procedimento de treinamento dos pesos. Geralmente, as arquiteturas de CNN mais bem sucedidas usam um grande número de dados rotulados para treinamento (e.g., ImageNet com 1,2 milhões de imagens divididas em 1000 classes). Por outro lado, um número pequeno de amostras faz com que o treinamento não convirja ou a rede treinada tenha resultados ruins. Quando se trata da Era de "Big Data" em que há uma enorme disponibilidade de dados, essa limitação pode ser facilmente solucionada. No entanto, alguns problemas podem não fornecer uma quantidade de dados razoável para um bom treinamento, por exemplo, tarefas relacionadas à biologia e medicina tendem a ter uma pequena quantidade de amostras devido à dificuldade de captura ou um número de pacientes pequeno.

As bases de dados de textura também possuem, em geral, menos de 100 amostras por classe. Assim, para tentar superar o número baixo de amostras, algumas técnicas de transferência de aprendizado têm sido utilizadas. A ideia da transferência de aprendizado é utilizar um modelo (i.e., os pesos) que já foi treinado em um conjunto de dados de um determinado domínio para

extrair características ou servir como ponto de partida para o treinamento de um novo conjunto de domínio distinto (ALOM *et al.*, 2018). A seguir, duas diferentes estratégias são descritas:

- **Extração de características:** nesta estratégia, um modelo de rede neural pré-treinado é utilizado como um extrator de características fixo. Para isso, basta escolher um modelo pré-treinado (por exemplo, na ImageNet) e remover as últimas camadas totalmente conectadas que tratam as saídas da rede (classificador). Em seguida, o restante da rede neural pode ser tratado como um extrator de características fixo para o novo problema. Por exemplo, se considerar as ativações ou saídas da camada oculta que está antes do classificador na arquitetura AlexNet, o vetor de características obtido terá dimensão de 4096 para cada imagem. Além disso, pode ser utilizado as saídas das camadas convolucionais, visto que algumas camadas extraem características específicas, como de texturas e borda. Assim, um classificador tradicional (e.g., SVM) pode ser treinado com os vetores de características obtidos de todas as amostras do novo problema.
- **Fine-Tuning:** esta estratégia é bastante empregada na maioria das aplicações utilizando redes neurais profundas. Basicamente, um modelo de CNN pré-treinado com uma base de dados grande é escolhido e, em seguida, utilizado como inicialização para o treinamento de uma tarefa específica. Nesse caso, um ajuste fino é realizado nos pesos de todas ou algumas camadas da rede neural pré-treinada para o novo domínio de interesse. Usualmente, as últimas camadas são escolhidas para reajuste dos pesos. Isso é motivado por observações de que as primeiras camadas das CNNs extraem características mais genéricas (e.g., detector de borda) que são importantes na maioria dos problemas e as últimas camadas contêm informações mais específicas do domínio.

Atualmente, estas estratégias têm sido utilizadas em diferentes problemas, tais como, na tarefa de detecção de objetos (RAZAVIAN *et al.*, 2014; REN *et al.*, 2015), em medicina (SAHA *et al.*, 2016; HOO-CHANG *et al.*, 2016) e identificação de doenças em folhas (MOHANTY; HUGHES; SALATHÉ, 2016). Em texturas, alguns trabalhos têm ganhado destaque através da aplicação do *Fisher Vector* nas saídas da última camada convolucional de um modelo pré-treinado (ANDREARCZYK; WHELAN, 2016; CIMPOI *et al.*, 2016). Outra forma é aplicar o *Global Average Pooling* (GAP) na última camada convolucional para converter os mapas de características em um vetor de características unidimensional (LIN; CHEN; YAN, 2013).

3.2.3 Redes Neurais Profundas em Grafos

Os métodos de aprendizado profundo provaram ser extremamente bem sucedidos em problemas de visão computacional em que os dados possuem uma estrutura de grade (BRUNA *et al.*, 2013). No entanto, as CNNs podem ser aplicadas apenas em dados representados por uma estrutura de baixa dimensão, ou seja, em grades 2D ou sequências 1D. De fato, os dados

podem ser naturalmente representados por grafos (ou redes complexas), como em engenharia de software (BARESI; HECKEL, 2002), proteômica (BALDI; POLLASTRI, 2004) e descrição de cena (KRAHMER; ERK; VERLEG, 2003). Nesse sentido, recentemente, visando se aproveitar do sucesso das CNNs, vários trabalhos têm explorado arquiteturas de aprendizado profundo para grafos (BRUNA *et al.*, 2013; LECUN; BENGIO; HINTON, 2015; NIEPERT; AHMED; KUTZKOV, 2016; SCARSELLI *et al.*, 2009; HENAFF; BRUNA; LECUN, 2015; ZHOU *et al.*, 2020).

Em geral, os métodos de aprendizado profundo para grafos estendem conceitos de técnicas de imagens para grafos, ou seja, estendem modelos de CNNs propostos originalmente para dados no domínio Euclidiano para o domínio não-Euclidiano (ZHOU *et al.*, 2020). Em técnicas de aprendizado profundo para imagens, um campo receptivo é movido sobre a imagem de acordo com um tamanho de passo, lendo os valores das intensidades dos pixels. Os pixels da imagem têm um arranjo espacial, portanto, o campo receptivo é movido da esquerda para a direita e de cima para baixo. Em grafos, não existe ordem nos vértices e estes não estão alinhados entre dois quaisquer grafos. Portanto, alguns trabalhos investigam uma sequência para os vértices da vizinhança e um mapeamento do grafo para uma representação vetorial ou matricial em que os vértices com vizinhança estrutural parecida sejam posicionados similarmente na nova representação (NIEPERT; AHMED; KUTZKOV, 2016). Para isso, algumas medidas de grafos são usadas para rotular os vértices, tais como, caminhos mais curtos e caminhadas aleatórias. Assim, é possível aprender representações de vizinhança com métodos de aprendizado profundo tradicionais.

Scarselli *et al.* (2009) propuseram um novo modelo de RNA que amplia os métodos existentes para aprender os dados modelados em grafos. Esse modelo ajudou a popularizar as redes neurais de grafos (em inglês, *Graph Neural Network* (GNN)), permitindo processar a maioria dos tipos de grafos (por exemplo, não direcionados, direcionados e cíclicos). O método implementa uma função que mapeia um grafo em um espaço euclidiano m -dimensional. Para aprendizado, um algoritmo supervisionado baseado em um mecanismo de difusão de informação (isto é, caminhadas aleatórias) é derivado para estimar os parâmetros do modelo proposto. Bruna *et al.* (2013) propôs uma generalização das convoluções para grafo através do grafo laplaciano. Esse método, chamado de construção espectral, é baseado nas propriedades de convoluções no domínio de Fourier. Essas construções permitem uma propagação eficiente e podem ser aplicadas em grafos grandes.

Um algoritmo, chamado *DeepWalk*, que aprende representações sociais dos vértices de um grafo através da modelagem do fluxo de caminhadas aleatórias foi proposto por Perozzi, Al-Rfou e Skiena (2014). As representações sociais são características latentes dos vértices que capturam a similaridade da vizinhança e codificam as relações sociais em um espaço vetorial. A técnica usa redes neurais para aprender uma linguagem especial composta por um conjunto de caminhadas geradas aleatoriamente. Portanto, no *DeepWalk*, dado um grafo como entrada, uma

representação latente é produzida como uma saída.

Recentemente, [Xin, Zhang e Shao \(2018\)](#) propuseram um *framework* de classificação de redes complexas integrando *embedding* e redes neurais convolucionais. Para tanto, os autores transformaram o grafo em um espaço de alta dimensão através do algoritmo *DeepWalk*, que preserva as estruturas locais do grafo e o converte em uma imagem bidimensional. Em seguida, as imagens foram utilizadas para treinar CNNs. Ao usar grafos sintéticos e dados reais de redes de comércio internacional, os resultados mostraram que o método pode classificar as categorias de redes com alta precisão e robustez. O trabalho desenvolvido por [Zhou *et al.* \(2020\)](#) apresenta uma revisão sistemática de métodos de aprendizado profundo com grafos para diferentes tarefas (e.g., classificação de vértices, predição de arestas etc.) e as suas possíveis aplicações.

MODELAGEM DE TEXTURAS EM REDES COMPLEXAS DIRECIONADAS

Um dos objetivos estabelecidos nesta tese refere-se ao estudo de abordagens para modelagem de imagens ou vídeos como redes complexas. Portanto, neste capítulo, abordagens para modelagem que mapeiam três categorias de texturas em redes complexas são apresentadas. Como já discutido na [Seção 2.4](#), em redes complexas, a etapa de modelagem é chave para a compreensão e caracterização dos dados ou sistemas de interesse. Nessa etapa, as principais propriedades e informações devem ser representadas, precisamente, nas redes complexas modeladas. Em análise de texturas, tanto em imagens como vídeos, a modelagem também deve capturar toda a sua complexidade. Contraditoriamente, embora texturas possam ser formadas por padrões básicos regulares, muitas delas são compostas de elementos com cor, brilho e forma que, quando vistos globalmente, formam padrões complexos. Isso faz com que não haja um consenso na comunidade científica para uma única definição formal de texturas, sendo essa motivada pela percepção particular ou muitas vezes associada ao problema estudado ([RIBAS, 2017](#)). No entanto, padrões complexos estão presentes, principalmente, em muitas texturas advindas da natureza, tal como em folhas. Nesse sentido, as redes complexas se apresentam como ferramentas poderosas devido a sua capacidade de modelar e analisar padrões complexos. Assim, elas podem ser consideradas uma medida de irregularidade que permite quantificar a complexidade das texturas.

Na última década, vários trabalhos na literatura ([CANTERO *et al.*, 2018](#); [GONÇALVES; MACHADO; BRUNO, 2015](#); [SCABINI *et al.*, 2019a](#)) têm empregado métodos baseados em redes complexas para análise e classificação de texturas. A maioria desses trabalhos considera cada pixel da imagem como um vértice da rede, com as conexões definidas com base na similaridade de intensidade e proximidade espacial dos pixels. Para tanto, em geral, quatro parâmetros de modelagem são utilizados: (i) um conjunto de raios, que definem a distância máxima para conexões, e (ii) um conjunto de limiares para remoção de arestas definido, usualmente, por três

valores (limiar inicial, de incremento e final). Em muitos casos, o ajuste desses parâmetros de modelagem se torna uma tarefa custosa devido às possibilidades de combinações de valores.

Para tentar superar essas limitações e aprimorar a precisão da modelagem, ao longo deste trabalho, novas formas de modelagem de texturas em redes complexas direcionadas foram estudadas. Além do bom desempenho em representar as principais propriedades de texturas por meio de redes complexas, essas abordagens utilizam, apenas, o raio como parâmetro de modelagem, eliminando a necessidade de encontrar valores de limiares ideais. Assim, os métodos desenvolvidos para análise de texturas, se diferenciam em: (i) utilizar apenas o parâmetro de raio de conexão para analisar a dinâmica de evolução da rede complexa; (ii) aplicar funções de pesos para as arestas direcionadas que mapeiam as informações de cor e distância entre os vértices; e (iii) como se trata de redes direcionadas ponderadas, usar medidas de grau de entrada e saída, e força de entrada e saída para compor o conjunto de características que irá treinar as redes neurais. O desempenho dessas abordagens é demonstrado no [Capítulo 5](#), que as utilizou em combinação com as RNNs para desenvolver métodos inovadores em análise de texturas.

Para o desenvolvimento de tais abordagens, contou-se com a colaboração de outros pesquisadores, como o estudante de doutorado Leonardo Felipe Santos Scabini, nos estudos de texturas em níveis de cinza e coloridas, do estudante de doutorado Rayner Harold Montes Condori, em relação à abordagem para texturas coloridas, dos professores Jarbas Joaci de Mesquita Sá Junior (estudos de texturas em níveis de cinza e dinâmicas) da Universidade Federal do Ceará, e Antoine Manzanera da École Nationale Supérieure de Techniques Avancées (ENSTA/IP de Paris), no trabalho de texturas dinâmicas. A seguir três abordagens para modelar três diferentes tipos de texturas como redes complexas direcionadas são apresentadas: (i) texturas em imagens em níveis de cinza – (ii) texturas em imagens coloridas RGB e – (iii) texturas dinâmicas em vídeos.

4.1 Modelagem de Texturas em Níveis de Cinza

Nesta seção, a abordagem estudada para modelagem de texturas em níveis de cinza como redes complexas direcionadas é descrita. Essa modelagem é parte do estudo publicado em [Ribas *et al.* \(2020\)](#), e foi adotada em combinação com as RNNs nos dois métodos propostos na [Seção 5.1](#). Primeiramente, considera-se uma imagem I composta por pixels i que tem como coordenadas Cartesianas x_i e y_i . Em imagens em níveis de cinza, cada pixel possui uma intensidade de nível de cinza representada por um valor inteiro $I(i) \in [0, L]$, em que L é a intensidade máxima na imagem, usualmente 255 para imagens de 8-bit. Para a modelagem da imagem de textura em uma rede complexa direcionada $G = (V, E)$, cada pixel $i \in I$ é mapeado como um vértice $v_i \in V$. O conjunto de arestas E é construído por meio da conexão de dois vértices v_i e v_j que representam dois pixels i e j com uma aresta direcionada de v_i para v_j , $e_{v_i, v_j} \in E$, se a distância Euclidiana entre os dois pixels é menor ou igual a um raio máximo de conexão r e a intensidade

de $I(i) \leq I(j)$,

$$E = \{e_{v_i, v_j} \in E \mid d(v_i, v_j) \leq r \wedge I(i) < I(j)\}, \quad (4.1)$$

em que $d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ é a distância Euclidiana entre os dois pixels. Na [Figura 7](#), exemplificam-se as vizinhanças para três raios diferentes. Ressalta-se que a direção da aresta é determinada pelas intensidades de nível de cinza dos pixels. Isto é, uma aresta irá apontar para o vértice que representa um pixel com a maior intensidade. Caso ambas intensidades sejam iguais, a aresta será bidirecional. Além disso, cada aresta possui um peso $w(e_{v_i, v_j})$ definido como uma combinação entre a diferença de intensidades e a posição espacial dos pixels,

$$w(e_{v_i, v_j}) = \begin{cases} \frac{|I(i) - I(j)|}{L}, & \text{se } d(v_i, v_j) = r \\ \frac{\left(\frac{d(v_i, v_j) - 1}{r - 1}\right) + \left(\frac{|I(i) - I(j)|}{L}\right)}{2}, & \text{caso contrário,} \end{cases} \quad (4.2)$$

em que $r = 1$ é a menor distância possível, assim, é considerado somente a informação de nível de cinza. Essa equação produz pesos de arestas normalizados no intervalo $[0, 1]$, dando importância igual à intensidade e à posição espacial dos pixels na imagem. Tal peso desempenha um papel importante na estrutura da rede, além de fornecer medidas como o grau ponderado dos vértices. É importante ressaltar que o valor r será o único parâmetro da modelagem e determinará o tamanho da vizinhança de cada vértice. Assim, ao modificar o parâmetro de raio r , é possível controlar a densidade da rede complexa, ou seja, à medida que r aumenta, o número de conexões e o grau dos vértices também aumentarão. Com um conjunto de raios $r = \{1, 2, \dots, R\}$, diferentes redes $G^1, G^2, \dots, G^R$ podem ser modeladas e utilizadas em uma análise da dinâmica de evolução do sistema (imagem). Tal evolução está relacionada à interação entre pixels vizinhos e, portanto, contém informações de textura multiescala. A [Figura 8](#) ilustra a estrutura da rede direcionada modelada usando diferentes valores de $r = \{1, 2, 3\}$. Observe que como as arestas estão apontando para a direção do gradiente, os vértices/pixels com maior intensidade tornam-se hubs locais se considerado o grau de entrada. Além das propriedades de grau, informações de diferença de intensidade e localização dos vértices também são incorporadas nas RCs através dos pesos das arestas. Isso faz com que as RCs geradas contenham relevantes informações topológicas que refletem as principais propriedades de textura tal como a relação entre elementos estruturais.

4.1.1 Complexidade Computacional

O custo da modelagem está diretamente associado à dois fatores: número de pixels da imagem $N = w * h$; e (ii) o raio máximo R usado para construir a vizinhança dos vértices/pixels. Por meio do raio máximo R é possível derivar as vizinhanças para os demais valores de raios usados $r = \{1, 2, \dots, R\}$. Assim, o custo para calcular a vizinhança de um pixel/vértice usando o

Figura 7 – Exemplo de vizinhança completa para três raios diferentes. Os pixels representados por esferas em verde serão conectados com o pixel central colorido em azul.

Fonte: Adaptada de [Scabini, Ribas e Bruno \(2020\)](#).

Figura 8 – Estrutura de redes complexas direcionadas modelando pixels com diferentes valores do parâmetro de modelado r .

Fonte: Adaptada de [Ribas et al. \(2020\)](#).

raio máximo é de $\frac{(2R+1)^2-1}{2}$ operações. Isso leva a uma complexidade de modelagem da imagem como rede complexa de $O(\frac{(2R+1)^2-1}{2}N)$.

4.2 Modelagem Espaço-Espectral de Texturas Coloridas

Esta seção apresenta a abordagem de modelagem de textura colorida em redes complexas espaço-espectrais direcionadas, publicada, inicialmente, no artigo em [Scabini, Ribas e Bruno \(2020\)](#) e também utilizada na metodologia proposta na [Seção 5.2](#). A maioria dos trabalhos da literatura não explora os padrões espectrais entre os canais de cores para caracterizar texturas coloridas, ou seja, analisa cada canal separadamente sem explorar a relação entre eles. Esta abordagem modela a relação dos pixels intra e inter-canais através de um esquema de rede complexa direcionado, denominado rede espaço-espectral ou *Spatio-Spectral Network* (SSN).

Considere uma imagem colorida I de $(w * h)$ pixels e z canais de cores com valores de pixels entre $[0, L]$. Essa imagem é modelada como uma rede complexa $G = (V, E)$ formada pelos conjuntos de vértices V e arestas E . Para tanto, cada vértice v_i mapeia um pixel i da imagem que está localizado em um determinado canal, formando o conjunto $V = \{v_1, \dots, v_{w*h*z}\}$. Cada vértice v_i possui duas coordenadas (x_i, y_i) mapeando a posição do pixel i na imagem, um índice z_i para o canal e o valor $I(v_i) \in [0, L]$ da intensidade do pixel em seu respectivo canal. Dadas essas definições, pode ser construída uma rede complexa multi-camadas em que cada camada representa um canal de cor.

Para criar conexões, um conjunto de raios $r = \{1, 2, \dots, R\}$ é utilizado para definir a vizinhança $\Omega_{v_i}^r$ do vértice v_i usando um critério de conexão radialmente simétrico:

$$\Omega_{v_i}^r = \{v_j \in V_N | r - 1 < d(v_i, v_j) \leq r\}, \quad (4.3)$$

em que $d(v_i, v_j)$ é a distância Euclidiana 2-D entre os vértices. Ou seja, dois vértices v_i e v_j estão conectados por uma aresta e_{v_i, v_j} se a distância $d(v_i, v_j)$ está em um intervalo entre o raio atual r e o raio anterior $(r - 1)$, portanto, $\{\Omega_{v_i}^1, \dots, \Omega_{v_i}^{r-1}\} \not\subseteq \Omega_{v_i}^r$. Note que como é considerado somente as coordenadas espaciais, os vértices são conectados entre si em todos os canais de cores. Para $\Omega_{v_i}^1$ os pixels estão conectados a si mesmos em canais distintos, ou seja, quando a distância é 0. A Figura 9 ilustra a vizinhança radialmente simétrica para um canal de cor usando diferentes raios de conexão. Cada aresta possui um peso com valor entre $[0, 1]$ calculado usando uma definição que equilibra a diferença de cor e distância entre os pixels (SCABINI; GONÇALVES; JR, 2015; SCABINI *et al.*, 2019a),

$$w(e_{v_i, v_j}) = \frac{(|I(v_i) - I(v_j)| + 1)(d(v_i, v_j) + 1) - 1}{(L + 1)(r + 1) - 1}. \quad (4.4)$$

Figura 9 – Vizinhança radialmente simétrica para três raios diferentes considerando apenas um canal. Os pixels representados por esferas em verde serão conectados com o pixel central colorido em azul.

Fonte: Adaptada de Scabini, Ribas e Bruno (2020).

Note que essas definições de vizinhança e peso realizam processamento de cor oponente de maneira espacial (SCABINI; RIBAS; BRUNO, 2020). As arestas e_{v_i, v_j} são direcionadas através da direção do gradiente, ou seja, o vértice v_i aponta para o vértice v_j se a sua intensidade de cor for menor, de modo que $I(v_i) \leq I(v_j)$. Assim, o único parâmetro de modelagem é o conjunto de raios $r = \{1, 2, \dots, R\}$ e o conjunto de arestas é definido por:

$$E = \{e_{v_i, v_j} \in E \mid r - 1 < d(v_i, v_j) \leq r \wedge I(v_i) < I(v_j)\}. \quad (4.5)$$

A aresta é bidirecional se os vértices tiverem a mesma intensidade de cor. O resultado dessa modelagem é uma rede complexa com propriedades topológicas que representam os padrões de textura colorida, que podem ser analisados e incorporados por medidas de vértice simples. Além disso, propriedades espaço-espectrais da rede são extraídas dos padrões de conexão entre canais, que são destacados por redes derivadas da rede original como em (SCABINI *et al.*, 2019a). Para tanto, duas redes complexas são derivadas: (i) espacial $G_S^r = (V_S, E_S)$ com apenas arestas entre o mesmo canal, tal como

$$\forall e(v_i, v_j) \in E, E_S = \{e(v_i, v_j) \mid z_i = z_j\}, \quad (4.6)$$

e – (ii) espectral $G_T^r(V_T, E_T)$ que contém um conjunto de arestas conectando vértices em canais diferentes,

$$\forall e(v_i, v_j) \in E, E_T = \{e(v_i, v_j) \mid z_i \neq z_j\}. \quad (4.7)$$

Desta forma, são obtidas três redes complexas diferentes G , G_S e G_T cujos conjuntos de vértices ($V = V_S = V_T$) são iguais, mas com conjuntos de arestas distintos. A Figura 10 ilustra a estrutura multi-camada dessas três redes modelando uma imagem de 3×3 pixels com um raio de conexão $r = 1$. É importante mencionar que a característica espaço-espectral da RC vem das conexões entre os canais que capturam a relação espectral dos pixels, enquanto as arestas entre pixels no mesmo canal definidas por uma vizinhança simétrica fornecem relevantes informações locais da textura. Adicionalmente, informações de gradiente da imagem também são incorporadas na rede através da direção das arestas e o seu peso. Essas propriedades permitem que a SSN modele diferentes e ricas características de texturas coloridas presentes na imagem (SCABINI; RIBAS; BRUNO, 2020).

4.2.1 Complexidade Computacional

Do mesmo modo que na modelagem em nível de cinza, o custo desta modelagem é em função do número de pixels da imagem ($N = w * h * z$) e do maior raio de conexão R considerado para construir a vizinhança dos vértices. No entanto, por se tratar de imagens RGB, são necessárias $\frac{(2R+1)^2-1}{2}z + z$ operações para construir a vizinhança de um pixel. É importante

Figura 10 – Estruturas multi-camadas das SSNs que modelam três canais de uma imagem 3×3 com raio $r = 1$.

Fonte: Adaptada de Scabini, Ribas e Bruno (2020).

notar que as redes G , G_S e G_T são construídas simultaneamente, uma vez que é necessário acessar os canais para verificar a existência de conexões. Portanto, a complexidade computacional desta modelagem é $O(N(\frac{(2R+1)^2-1}{2}z) + z)$.

4.3 Modelagem Espaço-Temporal de Texturas Dinâmicas

Uma extensão da modelagem de redes complexas direcionadas para texturas dinâmicas é apresentada nesta seção. A rede complexa gerada por esta modelagem é combinada com as RNNs para desenvolver o método de análise de textura dinâmica descrito na Subseção 5.3.2. Em muitos métodos existentes na literatura, a principal desvantagem é a forte ênfase somente em propriedades de aparência ou movimento do vídeo, com uma fraca combinação desses dois aspectos. Para aprimorar a modelagem e caracterização, na abordagem proposta, o vídeo de textura dinâmica é modelado em duas redes complexas direcionadas: (i) rede espacial $G_S = (V_S, E_S)$ que representa as propriedades de aparência e – (ii) rede temporal $G_T = (V_T, E_T)$ que mapeia as características de movimento. Cada pixel $i = (x_i, y_i, t_i)$ do vídeo é representado por um vértice $v_i \in V$, de modo que x_i e y_i são as coordenadas espaciais e t_i o índice temporal do pixel i . Além disso, a intensidade de nível de cinza dos pixels é dada por $I(i) \in [0, L]$, sendo L a intensidade máxima.

As duas redes possuem como principal diferença a definição do conjunto de arestas. Na rede espacial, para caracterizar a aparência, apenas vértices que estão no mesmo quadro

são conectados. Assim, o conjunto E_S é dado pela conexão de todos os pares de vértices cujas distâncias são menores ou iguais a um determinado raio r e as coordenadas de tempo t_i e t_j são iguais (como ilustrado na Figura 11 (a)),

$$E_S = \{e_{v_i, v_j} \in E_S \mid ((x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2) \leq r \wedge I(v_i) < I(v_j) \wedge t_i = t_j\}. \quad (4.8)$$

Por outro lado, na rede temporal, o foco é nas relações entre os quadros para analisar as características do movimento. Dessa maneira, no conjunto E_T , todos os vértices cujas distâncias são menores ou iguais à r e suas coordenadas de tempo diferem foram conectados (como exemplificado na Figura 11 (b)),

$$E_T = \{e_{v_i, v_j} \in E_T \mid ((x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2) \leq r \wedge I(v_i) < I(v_j) \wedge t_i \neq t_j\} \quad (4.9)$$

Cada pixel representado por um vértice possui um nível de cinza diferente, refletindo diretamente os padrões de textura. Para adicionar essas informações à topologia do grafo, ele foi transformado em direcionado e ponderado. Para isso, primeiramente, para cada aresta, uma direção é definida conforme os níveis de cinza dos pixels. Especificamente, em uma aresta e_{v_i, v_j} , o vértice v_i aponta para o vértice v_j se $I(i) \leq I(j)$. O peso $w(e_{v_i, v_j})$ é definido pela diferença de intensidades entre os dois pixels que representam os vértices:

$$w(e_{v_i, v_j}) = \begin{cases} \frac{|I(i) - I(j)|}{L}, & \text{se } r = 1 \\ \frac{\left(\frac{d(v_i, v_j) - 1}{r - 1}\right) + \left(\frac{|I(i) - I(j)|}{L}\right)}{2}, & \text{caso contrário} \end{cases} \quad (4.10)$$

em que $d(v_i, v_j)$ é a distância Euclidiana 3-D entre os pixels i e j . Note que quando o raio é $r = 1$, então o peso $w(e_{v_i, v_j})$ é a diferença dos níveis de cinza normalizados pelo nível de cinza máximo L , produzindo um valor em $[0, 1]$. Por outro lado, se $r > 1$ então o peso $w(e_{v_i, v_j})$ é determinado pela média da distância entre os pixels e a diferença da escala de cinza normalizada. Essa função de peso inclui informações sobre a vizinhança dos pixels e a diferença de nível de cinza, equilibrando a importância entre propriedades geométricas e cores na representação da textura dinâmica (GONÇALVES; MACHADO; BRUNO, 2015). Dessa forma, as duas redes complexas geradas mapeiam as principais propriedades de aparência e movimento das texturas dinâmicas.

Similarmente às modelagens das seções anteriores, nesta abordagem, o único parâmetro utilizado é o raio de conexão r . Portanto, pode-se empregar uma estratégia de análise da dinâmica de evolução da topologia das redes a partir de diferentes valores de raio $r = \{1, 2, \dots, R\}$. Essa abordagem difere-se dos trabalhos anteriores (GONÇALVES; MACHADO; BRUNO, 2015; RIBAS; GONÇALVES; BRUNO, 2019) por utilizar redes complexas direcionadas com: (i)

arestas ponderadas por uma função que equilibra a distância entre os pixels e a diferença de cor, ao invés de considerar somente a diferença de cor; e (ii) análise da dinâmica de evolução da topologia da RC através apenas do parâmetro de raio de conexão, enquanto os trabalhos anteriores usaram limiares para remoção de arestas.

Figura 11 – Modelagem de três quadros em redes complexas direcionadas usando raio $r = 1$.

(a) Rede espacial

(b) Rede temporal

Fonte: Ribas, Gonçalves e Bruno (2019), Ribas *et al.* (2022).

4.3.1 Complexidade Computacional

A complexidade para mapear vídeos de texturas dinâmicas em redes complexas é determinada pelo tamanho do vídeo de textura dinâmica e a vizinhança 3D. Em um vídeo, o número de pixels é $N = w * h * t$, em que w e h são respectivamente a largura e altura dos quadros e t o número de quadros. O número de operações para calcular as vizinhanças 3D é em função do raio

máximo R , sendo, assim, necessárias $((\frac{(2R+1)^3-1}{3}))$ operações. É importante ressaltar que a construção de ambas as redes complexas (espacial e temporal) pode ser realizada simultaneamente, pois é necessário verificar se a conexão é entre canais ou no mesmo canal. Assim, a complexidade computacional para calcular a vizinhança de todos os N pixels e, conseqüentemente, da abordagem de modelagem é $O(N(\frac{(2R+1)^3-1}{3}))$.

4.4 Considerações Finais

Neste capítulo, três técnicas de modelagem para mapear diferentes categorias de texturas em redes complexas direcionadas foram apresentadas. Inicialmente, a modelagem de texturas em níveis de cinza como redes complexas direcionadas é descrita, seguida das extensões para os casos de texturas coloridas e dinâmicas. As três técnicas compartilham das mesmas propriedades: um único parâmetro de modelagem que é o raio máximo e a direção das arestas determinada pela direção do gradiente. Essas duas propriedades permitem modelagens que, embora sejam simples e otimizadas, conseguem capturar e incorporar os principais padrões presentes nos diferentes tipos de texturas. Tal capacidade para modelar os padrões de texturas é verificada no [Capítulo 5](#), que propõe métodos inovadores de análise de texturas que combinam as modelagens apresentadas neste capítulo com as RNNs.

APRENDIZADO DE REPRESENTAÇÕES USANDO RNN

Neste capítulo, as abordagens de aprendizado de características ou representações para análise e classificação de diferentes categorias de texturas são descritas. Como apresentado no [Capítulo 4](#), as redes complexas são ferramentas poderosas capazes de capturar e representar padrões texturais, tornando-se uma medida de irregularidade que pode caracterizar a complexidade da textura. Nessa perspectiva, muitos trabalhos da literatura empregaram RCs para modelagem e caracterização de imagens e vídeos. No entanto, a maioria desses trabalhos utilizaram medidas tradicionais advindas da teoria de redes complexas, tais como, estatísticas da distribuição de grau, agrupamento, caminhadas etc. Apesar dos resultados promissores, uma caracterização mais sofisticada pode aprimorar a descrição dos padrões complexos presentes nas RCs em comparação com medidas simples e, assim, incrementar o desempenho de classificação.

Um dos objetivos deste trabalho foi a investigação e desenvolvimento de maneiras mais robustas de medir e caracterizar as redes complexas usando modelos de redes neurais artificiais. Na última década, o paradigma de aprendizado automático de representações tem sido amplamente utilizado em diversos problemas como uma alternativa para extração de características. Nesse caso, as redes neurais são empregadas para aprender as representações automaticamente com base em muitos tipos de dados, em contraste com métodos mais tradicionais que produzem características conhecidas como "feito à mão" ou "artesanal" (do inglês, *handcrafted*). Nos métodos *handcrafted*, as características são calculadas utilizando um modelo (por exemplo, matemático ou estatístico) ou técnica projetada manualmente visando descrever aspectos específicos do problema (e.g., em imagem podem ser borda e forma), e compensando acurácia e eficiência computacional ([NANNI; GHIDONI; BRAHNAM, 2017](#)). Além disso, em tais técnicas é possível uma compreensão dos mecanismos por trás do seu funcionamento, o que é muito importante e necessário em muitos problemas e aplicações. Por outro lado, no paradigma de aprendizado de representações, é utilizado as camadas neurais de modelos de redes neurais

como extratores de características que variam das mais básicas até as mais específicas de um dado problema. Para isso, essas camadas são treinadas com grandes conjuntos de dados a fim de aprender automaticamente quais extratores de características (como de borda, textura, cor, forma etc) são mais apropriados para um determinado problema.

Com o objetivo de se aproveitar das melhores características de cada paradigma, nesta tese, a combinação de redes complexas e redes neurais randomizadas foi investigada. Enquanto as redes complexas se destacam pela capacidade de modelar e representar padrões, as RNNs são caracterizadas pela simplicidade, algoritmo de aprendizado rápido e bom desempenho com poucos dados para treinamento. As abordagens desenvolvidas combinam uma descrição baseada em uma modelagem de redes complexas, e representações aprendidas por RNNs como uma forma mais eficiente de caracterização. Para tanto, dada uma amostra de textura (em imagem ou vídeo), esta é modelada em rede complexa e, em seguida, informações topológicas são usadas para aprender a representação, que será formada pelos pesos da camada de saída da RNN treinada. É importante enfatizar que as RNNs são treinadas com informações de uma única amostra. Em outras palavras, como em uma abordagem *node2vec*, para cada vértice, um vetor de atributos e rótulo são obtidos para compor o conjunto de treinamento para as RNNs. Os passos descritos acima são ilustrados na Figura 12.

Figura 12 – Diagrama geral do processo de aprendizado de representações de redes complexas empregado nos métodos desenvolvidos neste capítulo. (a) Uma única amostra é modelada como rede complexa e, para cada vértice, um vetor de atributos é construído e um rótulo é atribuído. (b) Todos os vetores de atributos dos vértices da rede complexa formam a matriz de entrada que treinará a RNN. (c) Os pesos da camada de saída da RNN treinada serão utilizados para compor a representação final em (d).

Fonte: Elaborada pelo autor.

As abordagens desenvolvidas apresentadas neste capítulo utilizam as modelagens descritas no Capítulo 4 para diferentes categorias de texturas. Assim, no âmbito teórico, este capítulo apresenta as seguintes pesquisas. Na Seção 5.1, duas abordagens que combinam redes complexas e RNNs para caracterização de texturas em níveis de cinza são apresentadas. Esses trabalhos foram realizados em colaboração com o doutorando Leonardo F. S. Scabini e o prof. Jarbas J. M. Sá Junior. Na Seção 5.2, apresenta-se um estudo de aprendizado de representação espaço-

espectral para analisar texturas coloridas, desenvolvida em colaboração com os estudantes de doutorado Rayner H. M. Condori e Leonardo F. S. Scabini. Por fim, duas abordagens para texturas dinâmicas baseadas em RNNs e redes complexas são descritas na [Seção 5.3](#). Nesse caso, em um primeiro momento, investigou-se a aplicação direta das RNNs nos vídeos de texturas dinâmicas, para, posteriormente, combinar a descrição com redes complexas e RNNs, resultando em uma segunda abordagem. Essas duas abordagens também contaram com a colaboração dos professores Jarbas J. M. Sá Junior e Antoine Manzanera. No âmbito prático, no [Capítulo 8](#), alguns métodos desenvolvidos neste capítulo também são aplicados e avaliados em diversos problemas reais.

5.1 Aprendizado de Representações de Redes Complexas para Análise de Texturas em Níveis de Cinza

Texturas são padrões visuais formados pela variação e organização de pixels na imagem, estando diretamente relacionadas com a superfície de objetos. Devido às aplicações, muitas das técnicas para caracterização são monocromáticas, ou seja, consideram somente a luminância da imagem (nível de cinza). Ao longo de décadas, atributos de texturas têm sido empregados para caracterizar imagens em muitos problemas de visão computacional e processamento de imagens, tal como reconhecimento de plantas, análise de imagens médicas, inspeção industrial, análise de imagens microscópicas etc. Isso resulta em uma extensa e heterogênea literatura de métodos para descrição de texturas, que são empregados em diversas áreas de modo a beneficiar sistemas de reconhecimento de imagens. As técnicas computacionais clássicas para descrição de texturas podem ser agrupadas em quatro diferentes categorias: estatísticos ([OJALA; PIETIKAINEN; MAENPAA, 2002](#); [HARALICK, 1979](#)), espectrais ([MANJUNATH; MA, 1996](#); [VES *et al.*, 2014](#)), estruturais ([LAM; LI, 1997](#)) e baseado em modelos ([BACKES; CASANOVA; BRUNO, 2009b](#); [FLORINDO; BRUNO, 2012](#); [PANJWANI; HEALEY, 1995](#); [BACKES; CASANOVA; BRUNO, 2013](#)). Além dos clássicos, mais recentemente, técnicas inovadoras estão abordando a textura de forma diferente, alcançando resultados promissores. Um exemplo são as técnicas baseadas em aprendizado tal como redes neurais artificiais ([ZHANG *et al.*, 2017](#); [JUNIOR; BACKES, 2016](#)).

Nesta seção, são apresentadas duas abordagens para análise de texturas em níveis de cinza usando aprendizado de representações de redes complexas. Basicamente as duas abordagens compartilham a mesma ideia: (i) modelar as texturas em níveis de cinza em redes complexas direcionadas usando o modelo descrito na [Seção 4.1](#); (ii) usar propriedades topológicas das redes modeladas para aprender as características com RNNs, usando a camada de saída treinada para formar a representação. Assim, a principal diferença nas abordagens é na forma como as representações são aprendidas a partir das redes complexas. A primeira abordagem ([Subseção 5.1.1](#)) aprende as características a partir da dinâmica de evolução das redes complexas modeladas com

diferentes valores de raio r , enquanto na segunda abordagem, descrita na [Subseção 5.1.2](#), as representações são aprendidas empregando uma análise dos padrões complexos locais (isto é, da vizinhança dos pixels/vértices) das redes complexas. Em ambas as abordagens, a modelagem proposta tem como vantagem o uso de apenas um parâmetro (raio), enquanto as representações propostas baseadas em RNN aumenta o desempenho de classificação. Por fim, a [Subseção 5.1.4](#) e [Subseção 5.1.3](#) apresentam os experimentos, análises, resultados e discussões sobre os métodos desenvolvidos e da literatura.

5.1.1 Abordagem 1: Aprendizado de Representações da Evolução de Redes Complexas (ERCNN)

Esta abordagem proposta e os resultados estão descritos no artigo publicado no periódico *Pattern Recognition* ([RIBAS et al., 2020](#)). Considerando redes complexas direcionadas mapeando uma textura em níveis de cinza através da modelagem descrita na [Seção 4.1](#), esta seção apresenta uma abordagem para a análise de textura. Para tanto, tal abordagem consiste em empregar RNNs para o aprendizado de representações a partir da dinâmica de evolução das redes complexas construídas com diferentes valores de raio r . Esta abordagem é denominada nesta tese como aprendizado de representações da Evolução de Redes Complexas com RNN (ERCNN) para análise de texturas em níveis de cinza. É importante enfatizar que nessa abordagem, a RNN é treinada usando como entrada e saída características de uma única amostra de textura. Assim, para cada vértice é construído um vetor de atributos (que também pode ser entendido como uma abordagem *node2vec*) para compor a matriz de entrada e a intensidade de nível de cinza é utilizado como o rótulo. Dessa forma, o vetor de características ou representação da textura será composto pelos pesos treinados da camada de saída da RNN.

Representação Proposta

Inicialmente, da rede complexa modelada foram consideradas três informações para cada vértice: grau de saída, grau de saída ponderado e grau de entrada ponderado. Como o grau de saída está estritamente relacionado com o grau de entrada nas redes modeladas (i.e., a soma dos dois graus produz o mesmo valor em todos os vértices) e, portanto, possuem a mesma informação, somente os graus de saída foram considerados. O grau de saída $\bar{k}_{v_i}$ de um vértice v_i , representa o número de arestas de saída conectadas a outros vértices,

$$\bar{k}_{v_i} = \sum_{v_j \in V} \begin{cases} 1, & e_{v_i, v_j} \in E \\ 0, & \text{caso contrário.} \end{cases} \quad (5.1)$$

Por outro lado, o grau de saída ponderado ou força de saída $\bar{s}_{v_i}$ é dado pela soma dos pesos das arestas de saída de um vértice v_i ,

$$\bar{s}_{v_i} = \sum_{v_j \in V} \begin{cases} w(e_{v_i, v_j}), & e_{v_i, v_j} \in E \\ 0, & \text{caso contrário.} \end{cases} \quad (5.2)$$

Por fim, o grau de entrada ponderado ou força de entrada $\hat{s}_{v_i}$ é definido como a soma dos pesos das arestas de entrada em v_i ,

$$\hat{s}_{v_i} = \sum_{v_j \in V} \begin{cases} w(e_{v_j, v_i}), & e_{v_j, v_i} \in E \\ 0, & \text{caso contrário.} \end{cases} \quad (5.3)$$

Para construir a matriz de vetores de entrada para a RNN, uma estratégia de análise da dinâmica de evolução da topologia da RC para diferentes valores do parâmetro de modelagem r é adotada. Desta forma, o vetor de entrada e a saída desejada de um vértice v_i é construído do seguinte modo: o valor da intensidade de nível de cinza do pixel é considerado a saída desejada $y_{v_i} = I(i)$ e os valores de grau de saída do vértice para diferentes valores do parâmetro r são utilizados como atributos do vetor de entrada $\vec{x}_{v_i} = [\bar{k}_{v_i}^1, \bar{k}_{v_i}^2, \dots, \bar{k}_{v_i}^R]$, em que R é o número de raios usados. Uma matriz de vetores de entrada $X_{\bar{k}}$ para o grau de saída e uma matriz de saídas desejadas Y são então construídas considerando todos os vértices da rede complexa. Portanto, é possível analisar a dinâmica de evolução da topologia de vértices que representam pixels com uma determinada intensidade de nível de cinza. Observe que nesta abordagem todos os vetores de entrada da matriz e rótulos são derivados de um conjunto de redes complexas que representam uma única imagem de textura. Esse tipo de abordagem difere da adotada usualmente na área de aprendizado de máquina, em que cada vetor representa uma imagem. A [Figura 13](#) (a) ilustra a construção da matriz de entrada e vetor de rótulos a partir de redes complexas modelando uma textura usando vários valores de r . Além da matriz de vetores de entrada $X_{\bar{k}}$ contendo o grau de saída, matrizes de vetores de entrada para a força de saída $X_{\bar{s}}$ e para a força de entrada $X_{\hat{s}}$ foram construídas.

O próximo passo é definir os pesos da matriz H da camada de entrada da RNN. De modo geral, esses pesos são definidos aleatoriamente em cada nova rede neural que será treinada. No entanto, como é desejado que o método forneça a mesma assinatura para a mesma imagem de textura, é necessário usar os mesmos valores na matriz H . Deste modo, seguindo a estratégia empregada em [Junior e Backes \(2016\)](#), o gerador congruente linear clássico (em inglês, *Linear Congruent Generator* (LCG)) ([PARK; MILLER, 1988](#)) é utilizado para obter os números pseudo-aleatórios uniformes que irão compor a matriz H , de acordo com

$$U(n+1) = (a * U(n) + b) \bmod c, \quad (5.4)$$

onde U é a sequência numérica aleatória e os valores de a , b e c são parâmetros. A sequência U tem comprimento $L = Q * (p + 1)$ sendo iniciada por $U(1) = L + 1$, os valores dos parâmetros são $a = L + 2$, $b = L + 3$ e $c = L^2$, conforme adotado por [Junior e Backes \(2016\)](#). Portanto, a matriz H é composta do vetor U dividido em Q segmentos de comprimento $p + 1$, em que

Figura 13 – Construção de um vetor de atributos de entrada e rótulo correspondente para o grau de saída $\bar{k}_{v_i}$ usando diferentes valores de r para modelar as redes complexas. (a) O grau de saída $\bar{k}_{v_i}$ do vértice v_i para diferentes raios r é usado para construir o vetor de entrada e a intensidade de cinza do pixel i é usada como rótulo de saída. (b) A RNN é treinada usando matriz $X_{\bar{k}}$ e Y construídos considerando todos os vértices, e os pesos treinados da camada de saída são usados como assinatura.

Fonte: Ribas *et al.* (2020).

p é o número de atributos. Finalmente, todos os valores das matrizes H e as linhas de X são normalizados usando média zero e variação unitária (*z-score*).

A representação proposta é construída baseado na matriz f , que se torna um vetor $\vec{f} = YZ^T(ZZ^T + \lambda I)^{-1}$, em que $\lambda = 10^{-3}$ (Figura 13 (b)). Observe que $\vec{f}$ tem comprimento $Q + 1$ devido ao valor do bias. Portanto, primeiro é realizado a concatenação de vetores $\vec{f}$ obtidos para RNNs treinadas com as três matrizes de vetores de entrada $X_{\bar{k}}, X_{\bar{s}}, X_{\bar{s}}$, gerando o seguinte vetor:

$$\vec{Y}(Q)_R = [\vec{f}_{\bar{k}}, \vec{f}_{\bar{s}}, \vec{f}_{\bar{s}}], \quad (5.5)$$

em que Q é o número de neurônios da camada escondida e R é o raio máximo utilizado na modelagem da rede complexa, ou seja, o número de raios usados para construir a matriz X .

O vetor $\vec{Y}(Q)_R$ é construído com um único valor de Q e R . Esses dois parâmetros influenciam nos pesos da rede neural e, portanto, fornecem diferentes características para diferentes valores. Assim, inicialmente, o vetor $\vec{\Theta}(R)_{(Q_1, Q_2, Q_m)}$ que concatena os vetores $\vec{Y}(Q)_R$ para diferentes valores de Q é proposto,

$$\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m} = [\vec{\Upsilon}(Q_1)_R, \vec{\Upsilon}(Q_2)_R, \dots, \vec{\Upsilon}(Q_m)_R]. \quad (5.6)$$

Por fim, a representação de textura $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ que combina vetores $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ para dois valores de R distintos é proposta,

$$\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m} = [\vec{\Theta}(R_1)_{Q_1, Q_2, \dots, Q_m}, \vec{\Theta}(R_2)_{Q_1, Q_2, \dots, Q_m}]. \quad (5.7)$$

Estudo de Parâmetros

Para estudar o método desenvolvido, o impacto dos parâmetros R e Q e das diferentes representações propostas na tarefa de classificação de texturas em níveis de cinza é analisado. Para esse propósito, o desempenho do método através da métrica de acurácia usando quatro bases de dados é avaliado: Outex (OJALA *et al.*, 2002), USPTex (BACKES; CASANOVA; BRUNO, 2012), Brodatz (BRODATZ, 1966) e Vistex (PICARD *et al.*, 1995). A classificação foi realizada com o método *Linear Discriminant Analysis* (LDA) (EVERITT; DUNN, 2001) e esquema de validação *leave-one-out*. As bases de dados, classificadores e esquema de validação são descritos em detalhes no Apêndice A.

A Figura 14 mostra as acurácias obtidas nas bases de dados com o vetor de características $\vec{\Upsilon}(Q)_R$ usando $R = 4$. Neste experimento, seis valores diferentes de Q são testados $Q \in \{04, 09, 14, 19, 29, 39\}$. Como pode ser observado no gráfico, as taxas de acerto aumentam à medida que é incrementado o valor de Q . Esse incremento também é seguido por um aumento na dimensão do vetor de características usado. As melhores acurácias são obtidas usando $Q = 14$ na base de dados Vistex e $Q = 19$ nas demais. Esses valores de Q produzem representações de tamanho 45 ($Q = 14$) e 60 ($Q = 19$). Além disso, a taxa de acurácia se estabiliza quando é usado valores de Q maiores do que 14 na base Vistex e valores maiores do que 19 nas demais bases.

Figura 14 – Acurácias para a assinatura $\vec{\Upsilon}(Q)_R$ usando diferentes valores de Q nas quatro bases de dados.

Fonte: Ribas *et al.* (2020).

A Tabela 1 mostra as acurácias alcançadas nas quatro bases de dados usando a representação $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ com $R = 4$. Os resultados mostram que conforme os valores de Q e suas combinações aumentam (ou seja, o número de atributos aumenta), as taxas de sucesso também aumentam. Porém, vetores de características grandes não garantem o melhor desempenho, uma vez que as taxas de sucesso tendem a se estabilizar em um determinado valor. Por exemplo, se comparado o vetor $\vec{\Theta}(04)_{19,29,39}$, que tem 270 atributos, com o vetor $\vec{\Theta}(04)_{04,09,14}$, que possui 90 atributos, o primeiro apresenta desempenho inferior em todas as bases de dados, exceto na Brodatz. Isso sugere que a assinatura proposta atinge seu limite em termos de discriminação. Assim, os vetores $\vec{\Theta}(04)_{04,09,14}$ e $\vec{\Theta}(04)_{04,14,19}$ são considerados, pois apresentaram um bom equilíbrio entre alta acurácia e um pequeno número de atributos.

Tabela 1 – Resultados de classificação para o vetor $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ usando diferentes valores de Q e suas combinações para o raio máximo $R = 4$.

$\{Q_1, Q_2, \dots, Q_m\}$	# de Atributos	Outex	USPTex	Brodatz	Vistex
{04, 09}	45	88,60	94,07	93,02	97,22
{04, 14}	60	88,82	94,98	93,86	98,50
{04, 19}	75	88,97	95,46	94,88	97,92
{04, 29}	105	88,38	94,76	94,48	97,80
{04, 39}	135	87,57	95,42	95,27	98,15
{09, 14}	75	88,09	94,72	93,52	97,92
{09, 19}	90	88,60	94,42	94,26	98,15
{09, 29}	120	87,50	94,24	94,14	97,80
{09, 39}	150	86,76	94,81	94,88	97,80
{14, 19}	105	89,04	94,94	94,54	98,26
{14, 29}	135	88,09	94,90	94,48	97,92
{14, 39}	165	87,65	95,29	95,33	98,15
{19, 29}	150	87,50	94,59	94,76	97,45
{19, 39}	180	87,28	95,03	95,27	98,03
{29, 39}	210	85,88	94,37	95,05	97,92
{04, 09, 14}	90	89,34	95,46	95,05	98,61
{04, 09, 19}	105	89,71	95,55	95,16	98,26
{04, 09, 29}	135	88,68	95,51	94,88	97,80
{04, 09, 39}	165	87,94	95,90	95,72	97,80
{04, 14, 19}	120	89,41	95,94	95,21	98,38
{04, 14, 29}	150	88,68	95,59	95,05	98,26
{04, 14, 39}	180	88,53	95,90	95,89	98,84
{04, 19, 29}	165	89,12	95,94	95,27	98,03
{04, 19, 39}	195	88,68	95,90	95,61	98,38
{04, 29, 39}	225	87,94	95,03	95,72	98,38
{09, 14, 19}	135	89,56	95,24	94,99	98,50
{09, 14, 29}	165	88,75	95,42	95,10	98,15
{09, 14, 39}	195	88,01	95,68	95,61	98,73
{09, 19, 29}	180	88,75	95,24	94,93	98,26
{09, 19, 39}	210	87,94	95,38	95,05	97,92
{09, 29, 39}	240	88,01	94,59	94,99	97,92
{14, 19, 29}	195	88,75	95,33	95,05	98,50
{14, 19, 39}	225	88,38	95,72	95,44	98,61
{14, 29, 39}	255	88,09	95,03	95,50	98,50
{19, 29, 39}	270	88,01	94,63	95,50	97,92

Fonte: Ribas *et al.* (2020).

A representação $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ também é avaliada para diferentes valores de R . A Figura 15 apresenta as acurácias obtidas considerando as combinações $Q = \{04, 09, 14\}$ e $Q = \{04, 14, 19\}$. O valor máximo do raio R está associado à zona de conexão entre os pixels

(ou seja, vértices). Assim, valores menores de R modelam os pixels mais próximos e, à medida que R aumenta, o alcance da conexão também aumenta. Os resultados indicam que os menores valores de R proporcionam melhores acurácias quando comparados aos maiores valores. Isso sugere que os padrões locais são mais importantes do que os padrões globais para discriminar as texturas usadas neste trabalho. É importante ressaltar também que o valor de R determina o tamanho dos vetores de atributos dos vértices, por exemplo, para $R = 4$, o vetor de atributos de cada vértice será formado pelas medidas (grau ou força) usando quatro raios $\{1, 2, 3, 4\}$.

Figura 15 – Acurácias usando o vetor $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ para os dois melhores conjuntos de Q com diferentes valores de raio máximo R .

Fonte: Ribas *et al.* (2020).

Além disso, a combinação dos vetores $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ (mostrado na Tabela 1) com diferentes valores de raio máximo R foi estudada, resultando no vetor $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$. Para calcular esse vetor, as combinações de Q que forneceram os melhores resultados na Tabela 1 são utilizadas: $Q = \{4, 9, 14\}$ e $Q = \{4, 14, 19\}$. Neste experimento, o vetor $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ é calculado para dois valores de R (ou seja, até duas combinações de $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$) devido ao grande número de atributos gerados.

A Tabela 2 mostra os resultados dos vetores $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ usando a combinação $Q = \{04, 09, 14\}$. A maior acurácia foi fornecida pelo vetor $\vec{\Psi}(04, 06)_{04, 09, 14}$. Os resultados dos vetores $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ construídos com a combinação $Q = \{04, 14, 19\}$ são mostrados na Tabela 3. Nesse caso, os melhores resultados foram obtidos usando o vetor $\vec{\Psi}(04, 10)_{04, 14, 19}$. A Tabela 2 e Tabela 3 também mostram que combinando o vetor $\vec{\Theta}(R)_{Q_1, Q_2, \dots, Q_m}$ com diferentes valores de R , a acurácia aumenta cerca de 1% nas bases de dados. Entretanto, nos dois casos, as combinações de altos valores de R fornecem resultados inferiores. Além disso, embora os melhores resultados das duas tabelas ($\vec{\Psi}(04, 06)_{04, 09, 14}$ e $\vec{\Psi}(04, 10)_{04, 14, 19}$) sejam próximos, observe que o vetor $\vec{\Psi}(04, 10)_{04, 14, 19}$ tem um número de características maior que o vetor $\vec{\Psi}(04, 06)_{04, 09, 14}$. Portanto, as seguintes representações foram consideradas para este método: ERCNN¹ ($\vec{\Theta}(04)_{04, 09, 14}$) (que produz apenas 90 atributos), ERCNN² ($\vec{\Psi}(04, 06)_{04, 09, 14}$) e ERCNN³ ($\vec{\Psi}(04, 10)_{04, 14, 19}$).

Tabela 2 – Resultados de classificação considerando diferentes combinações de raios R usando $Q = \{4, 9, 14\}$

$\{R_1, R_2\}$	# de Atributos	Outex	USPTex	Brodatz	Vistex
{04, 06}	180	91,54	96,64	96,11	98,73
{04, 08}	180	91,47	96,25	95,88	98,26
{04, 10}	180	91,47	96,55	95,83	98,84
{04, 12}	180	91,69	96,25	95,72	98,26
{06, 08}	180	91,54	96,47	95,77	98,61
{06, 10}	180	90,74	96,25	95,83	98,50
{06, 12}	180	90,44	96,38	95,83	98,38
{08, 10}	180	90,58	96,03	95,15	98,49
{08, 12}	180	90,29	95,72	95,21	98,14
{10, 12}	180	90,59	95,46	94,93	98,38

Fonte: Ribas *et al.* (2020).Tabela 3 – Acurácias usando diferentes conjuntos de raio R com $Q = \{4, 14, 19\}$.

$\{R_1, R_2\}$	# de Atributos	Outex	USPTex	Brodatz	Vistex
{04, 06}	240	90,07	96,73	95,83	99,19
{04, 08}	240	91,17	96,68	96,05	98,95
{04, 10}	240	91,32	96,95	96,06	99,19
{04, 12}	240	91,76	96,64	96,11	98,61
{06, 08}	240	90,14	96,29	96,39	98,61
{06, 10}	240	89,63	96,60	95,95	98,50
{06, 12}	240	90,29	96,55	96,06	98,26
{08, 10}	240	90,00	95,94	95,72	98,37
{08, 12}	240	91,32	95,94	95,77	98,03
{10, 12}	240	90,51	96,03	95,05	98,03

Fonte: Ribas *et al.* (2020).

5.1.2 Abordagem 2: Aprendizado de Representações usando Padrões Locais Complexos (PLCNN)

Nesta segunda abordagem desenvolvida, a ideia também é caracterizar a topologia das redes complexas modeladas usando RNNs como na [Subseção 5.1.1](#), entretanto, o foco é na análise dos padrões locais dos pixels/vértices. Para tanto, propõe-se uma maneira diferente para construir os vetores de entrada dos vértices e seus respectivos rótulos a partir das redes complexas. Essa proposta aplica janelas de tamanho 3×3 na RC para construir a matriz de vetores de atributos de entrada para a rede neural. Nesta tese, esta abordagem é denominada como aprendizado de representações de Padrões Locais Complexos usando RNN (PLCNN) para análise de texturas em níveis de cinza. A seguir a representação proposta é detalhada.

Representação Proposta

Primeiramente, considere que as coordenadas cartesianas x_i e y_i de um vértice v_i são iguais ao do pixel i representado pelo vértice. Assim, a janela é a vizinhança espacial de um vértice com base nessas coordenadas cartesianas. Em outras palavras, divide-se a imagem em janelas conjuntas 3×3 e, em vez de usar as informações diretamente dos pixels como em trabalhos clássicos de texturas, são consideradas as medidas dos vértices que os representam (ou seja, da RC). A Figura 16 ilustra uma janela com um pixel central i representado pelo vértice v_i e pixels vizinhos representados por seus vértices. Em relação a essa janela, para cada vértice v_i , o grau de saída $\bar{k}_{v_i}$ é considerado o rótulo e os graus de saída dos vértices vizinhos compõem o seu vetor de atributos $\vec{x}_{v_i}$ da seguinte forma: $\vec{x}_{v_i} = [\bar{k}_{v_1}, \bar{k}_{v_2}, \bar{k}_{v_3}, \dots, \bar{k}_{v_8}]$.

Figura 16 – Vizinhança de um vértice/pixel.

v_1	v_2	v_3
v_8	v_i	v_4
v_7	v_6	v_5

Fonte: Elaborada pelo autor.

Nesta abordagem proposta, é possível aprender a representação usando um valor único de raio r para modelagem. Por outro lado, no método anterior (Subseção 5.1.1), é necessário utilizar um conjunto de valores de raio para analisar a dinâmica de evolução das RCs modeladas e então aprender a representação. Além disso, nesta proposta, ao invés de usar o nível de cinza, é adotado o grau de saída/entrada como rótulo. Portanto, os vetores de entrada e os rótulos são compostos apenas de informações da rede complexa.

A Figura 17 (a) ilustra como o vetor de atributos de entrada e seu respectivo rótulo são obtidos de um determinado vértice usando uma janela 3×3 . Uma matriz de vetores de atributos de entrada $X_{\vec{k}}$ e um vetor de rótulos Y para o grau de saída são então criados considerando todos os vértices da RC. Dessa forma, pode-se analisar os padrões complexos locais dos vértices que representam os pixels. Além da matriz de entrada $X_{\vec{k}}$ contendo o grau de saída, também são construídas matrizes usando a força de saída $X_{\vec{s}}$ e a força de entrada $X_{\vec{e}}$. O grau de saída é utilizado como rótulo para as matrizes de grau e força de saída, enquanto o grau de entrada é o rótulo da matriz de força de entrada.

Para definir a matriz de peso H das RNNs usadas no treinamento, o mesmo procedimento adotado na Subseção 5.1.1 foi empregado. Assim, o gerador LCG é utilizado para gerar a sequência de pesos aleatórios que irão compor a matriz de pesos H . Essa matriz H e as matrizes X (cada linha) também são normalizados com z -score.

Figura 17 – Esquema do método PCL para construir o vetor de atributos e rótulo a partir de cada vértice.

Fonte: Elaborada pelo autor.

As características da RC modelada e, consequentemente, a representação da textura são obtidos a partir da matriz f , que é composta pelos pesos aprendidos da camada de saída. Essa se torna um vetor $\vec{f} = YZ^T(ZZ^T + \lambda I)^{-1}$, onde $\lambda = 10^{-3}$, conforme ilustrado na Figura 17 (b). Assim, a primeira representação para uma textura é produzida pela concatenação dos vetores $\vec{f}$ de três RNNs treinadas com as três matrizes de vetores de atributos de entrada $X_{\bar{k}}, X_{\bar{s}}, X_{\bar{\delta}}$,

$$\vec{Y}(Q)_r = [\vec{f}_{\bar{k}}, \vec{f}_{\bar{s}}, \vec{f}_{\bar{\delta}}], \quad (5.8)$$

em que Q é o número de neurônios ocultos e r é o raio usado para construir a RC. A representação $\vec{Y}(Q)_r$ é criada usando um valor único de Q e r . Como esses dois parâmetros alteram tanto a RC modelada como o treinamento da RNN, diferentes características podem ser aprendidas dependendo dos valores considerados. Aproveitando-se disso, propõe-se uma representação $\vec{\Theta}(R)_{(Q_1, Q_2, Q_m)}$, que consiste na combinação dos vetores $\vec{Y}(Q)_r$ para diferentes valores de raio r ,

$$\vec{\Theta}(Q)_{r_1, r_2, \dots, r_m} = [\vec{Y}(Q)_{r_1}, \vec{Y}(Q)_{r_2}, \dots, \vec{Y}(Q)_{r_m}]. \quad (5.9)$$

Uma representação final $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ que combina o vetor $\vec{\Theta}(Q)_{r_1, r_2, \dots, r_m}$ usando diferentes valores de Q é proposta, de acordo com

$$\vec{\Psi}_{Q_1, Q_2, \dots, Q_n} = [\vec{\Theta}(Q_1)_{r_1, r_2, \dots, r_m}, \vec{\Theta}(Q_2)_{r_1, r_2, \dots, r_m}, \dots, \vec{\Theta}(Q_n)_{r_1, r_2, \dots, r_m}]. \quad (5.10)$$

Estudo de Parâmetros

Para analisar o comportamento do método em termos de parâmetros, experimentos de classificação de texturas usando quatro bases de dados são realizados. Nos experimentos, o classificador supervisionado LDA seguido de uma validação cruzada *leave-one-out* foi empregado. Detalhes das bases de dados e classificadores são descritos no [Apêndice A](#).

A representação analisada é a do vetor $\vec{\Theta}(Q)_{r_1, r_2, \dots, r_m}$ com $Q = 4$. Para essa representação, diferentes valores de $r = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ e suas combinações são testados. Os resultados estão resumidos na [Figura 18](#), que apresenta a média das acurácias obtidas nas quatro bases de dados. Neste experimento, foram combinados no máximo dois valores de r devido ao grande número de características geradas para combinações maiores. Os resultados indicam que para um único valor de r (diagonal principal da matriz), o melhor resultado é alcançado com $r = 5$ (88,17%). No entanto, um único valor de r produz resultados inferiores ao da combinação de dois valores. Observe que os melhores resultados são alcançados ao combinar um valor baixo de r com um valor alto (o melhor resultado de 93,56% é alcançado para $r = \{2, 9\}$). Os valores de r determinam o raio da modelagem da RC; ou seja, para valores baixos de r , os pixels são conectados aos vizinhos mais próximos (análise local), por sua vez, para valores elevados de r , uma análise global da textura é realizada. Assim, espera-se que a combinação de características locais e globais proporcione resultados superiores.

Figura 18 – Acurácias da representação $\vec{\Theta}(Q)_{r_1, r_2, \dots, r_m}$ usando $Q = 4$ para diferentes valores de r nas quatro bases de dados.

Fonte: Elaborada pelo autor.

Um segundo experimento para avaliar a representação $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ foi realizado. Nesse caso, vários valores de $Q = \{04, 09, 14, 19, 24, 29\}$ foram usados para $r = \{2, 9\}$. A [Figura 19](#) mostra as acurácias com a assinatura $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ nas quatro bases de dados com diferentes

valores de Q e suas combinações. Note que os menores resultados estão nas diagonais principais, ou seja, quando um único valor de Q é considerado. Por outro lado, quando dois valores de Q são considerados, a acurácia aumenta. A Tabela 4 apresenta os resultados para a combinação dos três valores de Q . Observe que maiores acurácias foram produzidas com essas combinações. No entanto, à medida que valores maiores de Q são combinados, o tamanho do vetor de características também aumenta e a acurácia tende a diminuir ou se estabilizar. Assim, foi considerada a representação $\vec{\Psi}_{04,19,29}$, que apresenta um bom equilíbrio entre acurácia e número de características nas quatro bases de dados.

Figura 19 – Resultados de classificação para a representação $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ com diferentes valores de Q .

Fonte: Elaborada pelo autor.

5.1.3 Análise das Representações Propostas

Para compreender as representações aprendidas pelos métodos propostos e a relação com a imagem, primeiramente, são analisadas as propriedades topológicas das redes complexas direcionadas geradas pela modelagem desenvolvida, que é compartilhada por ambas as abordagens propostas. A Figura 20 mostra as representações visuais das três medidas dos vértices obtidas

Tabela 4 – Resultados para o vetor $\vec{\Psi}_{Q_1, Q_2, \dots, Q_n}$ combinando três valores diferentes de Q .

$\{Q_1, Q_2, Q_3\}$	# de Atributos	Outex	Brodatz	USPTex	Vistex
{04, 09, 14}	180	91,69	96,45	96,29	99,77
{04, 09, 19}	210	91,76	97,41	97,03	100,00
{04, 09, 24}	240	91,25	97,75	96,77	99,88
{04, 09, 29}	270	91,10	98,20	97,12	99,88
{04, 14, 19}	240	91,76	97,52	96,90	99,88
{04, 14, 24}	270	91,99	98,03	96,95	99,77
{04, 14, 29}	300	91,10	98,37	96,86	99,77
{04, 19, 24}	300	92,43	97,97	97,08	99,77
{04, 19, 29}	330	92,13	98,09	97,21	99,88
{04, 24, 29}	360	90,96	97,97	96,99	99,65
{09, 14, 19}	270	91,91	97,47	97,16	100,00
{09, 14, 24}	300	91,25	97,80	97,21	99,77
{09, 14, 29}	330	91,10	98,31	97,21	100,00
{09, 19, 24}	330	91,47	97,92	97,16	99,77
{09, 19, 29}	360	91,84	98,25	97,21	99,88
{09, 24, 29}	390	90,74	97,97	97,03	99,77
{14, 19, 24}	360	91,91	97,92	97,25	99,77
{14, 19, 29}	390	91,99	98,03	97,29	99,88
{14, 24, 29}	420	91,18	97,92	97,21	99,77
{19, 24, 29}	450	91,10	97,86	97,03	99,54

Fonte: Elaborada pelo autor.

pela conversão de seu valor em um pixel em nível cinza. De maneira geral, é possível perceber que cada medida destaca diferentes padrões da variação de intensidade da imagem de forma multi-escala ao variar o raio r , sendo possível combinar tanto as diferentes medidas bem como os raios a fim de produzir um descritor eficiente. Por exemplo, para valores baixos de r (coluna 1), detalhes mais finos da textura são evidenciados pela RC e, à medida que se aumenta o raio r (coluna 4), esta tende a destacar características mais globais. Isso demonstra que o esquema de modelagem proposto captura as principais propriedades físicas da textura.

Na segunda etapa dos métodos, uma rede neural randomizada é usada para aprender a representação a partir das redes complexas modeladas. A representação $\vec{f}_k$ calculada para diferentes amostras de texturas usando a abordagem ERCNN é mostrado na [Figura 21](#). Essa representação é composta por pesos treinados da camada de saída e bias da RNN. Como pode ser visto, cada classe de textura possui uma assinatura diferente. Isso corrobora que as RNNs podem efetivamente aprender a sintetizar as medidas topológicas das redes complexas em informações de textura. Portanto, seus pesos aprendidos são capazes de representar e separar diferentes classes de textura.

Em relação ao método PCLNN, os pesos aprendidos da última camada para várias amostras de texturas são ilustrados na [Figura 22](#). Na figura, os vetores de pesos ou representações das amostras são representados pelas colunas e as linhas correspondem a cada peso ligando a camada

Figura 20 – Uma imagem de textura (a) da base de dados USPTex é modelada como uma rede complexa direcionada para diferentes valores de r e representações visuais das medidas de cada vértice são mostradas (d)-(f). Os valores dos pixels são obtidos normalizando cada medida de forma que o grau máximo do vértice para cada raio (4,60,180 e 364) corresponda à intensidade 255 (branco).

Fonte: Ribas *et al.* (2020).

oculta e a de saída. O valor do peso em $[0, 1]$ é representado em escala de cinza. Por meio dessa visualização, é possível notar que as amostras da mesma classe possuem representações semelhantes, enquanto amostras de classes diferentes possuem representações distintas. Por exemplo, no peso 29 é possível separar as classes c42 e c48, já os pesos 16 e 17 combinados podem separar as três classes. Assim, pode-se afirmar que as representações aprendidas conseguem descrever as imagens de acordo com seus padrões texturais.

Figura 21 – Amostras de textura (a) da base de dados Brodatz e seus respectivos vetores de características $\vec{f}_k$ em (b). Este vetor de característica foi calculado usando os parâmetros $R = 04$ e $Q = 04$.

Fonte: Ribas *et al.* (2020).

5.1.4 Resultados e Discussões

Esta seção apresenta os experimentos, resultados, comparações e discussões relacionadas as duas abordagens desenvolvidas combinando redes complexas direcionadas e RNNs para análise de texturas em níveis de cinza.

Configuração Experimental

Ambas as abordagens desenvolvidas são usadas para calcular as representações das imagens de texturas, assim como outros métodos da literatura. As representações propostas empregadas nos experimentos são as discutidas no estudo de parâmetros realizado para cada abordagem na sua respectiva seção. A configuração experimental utilizada foi a mesma para todos os métodos (LDA com um esquema de validação cruzada *leave-one-out*), exceto para o CLBP, que utilizou o classificador 1-*Nearest Neighborhood* (1NN) com distância chi-quadrada, conforme o artigo original. Nas comparações, a métrica de desempenho utilizada é a acurácia, além do número de atributos dos métodos. Mais detalhes sobre as métricas, classificadores e bases de dados estão disponíveis no Apêndice A. Os seguintes métodos da literatura são considerados nas comparações: *Grey-Level Co-occurrence Matrix* (GLCM) (HARALICK, 1979), *Gray Level Difference Matrix* (GLDM) (WESZKA; DYER; ROSENFELD, 1976), Fourier (WESZKA; DYER; ROSENFELD, 1976), Filtros de Gabor (MANJUNATH; MA, 1996; DAUGMAN; DOWNING, 1995), Fractal (BACKES; CASANOVA; BRUNO, 2009b), Fractal Fourier (FLORINDO; BRUNO, 2012), *Local Binary Pattern* (LBP) (OJALA; PIETIKAINEN; MAENPAA, 2002), *Local Binary Patterns Variance* (LBPV) (GUO; ZHANG; ZHANG, 2010b), *Complete Local Binary Pattern* (CLBP) (GUO; ZHANG; ZHANG, 2010a), *Local Phase Quantization* (LPQ) (OJANSIVU; HEIKKILÄ, 2008), *Local Configuration Pattern* (LCP) (GUO; ZHAO; PIETIKÄI-

Figura 22 – Vetores de pesos da camada de saída gerados pela abordagem PLCNN de amostras de textura de várias classes. Cada amostra é representada nas colunas e as linhas são os pesos.

Fonte: Elaborada pelo autor.

NEN, 2011), *Local Frequency Descriptor* (LFD) (MAANI; KALRA; YANG, 2013), *Binarized Statistical Image Features* (BSIF) (KANNALA; RAHTU, 2012), *Adaptive Hybrid Pattern* (AHP) (ZHU *et al.*, 2015), *Complex Network Texture Descriptor* (CNDT) (BACKES; CASANOVA; BRUNO, 2013) e assinatura ELM (JUNIOR; BACKES, 2016).

Adicionalmente, os métodos propostos também são comparados com modelos de redes neurais convolucionais profundas, que são aplicadas em uma abordagem de transferência de

aprendizado, em que uma arquitetura pré-treinada é usada como um extrator de características, calculando o *Global Average Pooling* (GAP) (LIN; CHEN; YAN, 2013) sobre a saída de sua última camada convolucional. Os pesos desses modelos são aprendidos no conjunto de dados ImageNet (DENG *et al.*, 2009), composto por milhões de imagens, e podem ser empregados para várias aplicações, como análise de textura (CIMPOI *et al.*, 2016). Foi considerado quatro modelos bem conhecidos da literatura: VGG19 (2014) (SIMONYAN; ZISSERMAN, 2014), InceptionV3 (2016) (SZEGEDY *et al.*, 2016), ResNet50 (2016) (HE *et al.*, 2016) e InceptionResNetV2 (2017) (SZEGEDY *et al.*, 2017). Todos os modelos e seus pesos pré-treinados foram importados da biblioteca Keras 2.2.4¹. Para esses métodos, as imagens são normalizadas pelo nível de cinza máximo possível antes de serem processadas.

Resultados Comparativos

Para validar as abordagens propostas para texturas em níveis de cinza, nomeadas como ERCNN (Subseção 5.1.1) e PLCNN (Subseção 5.1.2), os resultados obtidos pelos métodos são comparados com métodos da literatura. A Tabela 5 apresenta os resultados para todos os métodos nas quatro bases de dados de texturas avaliadas. Para a abordagem ERCNN, três representações diferentes foram consideradas: ERCNN¹ ($\vec{\Theta}(04)_{04,09,14}$), ERCNN² ($\vec{\Psi}(04,06)_{04,09,14}$) e ERCNN³ ($\vec{\Psi}(04,10)_{04,14,19}$). Quando comparadas as duas abordagens desenvolvidas, o método PLCNN consegue melhorar a acurácia em relação às representações ERCNN em todas as bases. Isso evidencia que a análise local dos padrões complexos de texturas pode fornecer melhores atributos. Os resultados também mostram que as abordagens propostas obtiveram os melhores resultados quando comparado aos demais em três bases de dados (Outex, USPTex e Vistex).

Com relação aos métodos baseados em redes neurais convolucionais, os modelos InceptionV3 e InceptionResNetV2 superaram levemente os métodos propostos apenas na base de dados Brodatz. No entanto, observa-se que esses métodos não tiveram uma eficácia competitiva na base de dados Outex, que é caracterizada por amostras com iluminações diferentes. Além disso, esses modelos tendem a ser mais complexos e custosos computacionalmente, além de produzir vetores de características maiores. É também importante ressaltar que os métodos propostos atingiram acurácias maiores do que a assinatura ELM e o método CNDT, que são baseados em RNNs e redes complexas, respectivamente. Isso sugere que os métodos desenvolvidos obtiveram desempenhos superiores por apresentar simultaneamente as características principais de ambas as técnicas.

Análise de Rotação

A robustez à rotação é uma propriedade importante e desejável em problemas e aplicações que usam métodos de análise de textura. Portanto, experimentos foram realizados para avaliar a capacidade dos métodos em lidar com texturas rotacionadas. Nesse experimento, três bases

¹ <www.keras.io/applications>

Tabela 5 – Comparação das acurácias de diferentes métodos de análise de textura em quatro bases de dados usando o classificador LDA. Esses resultados foram reproduzidos para esta tese.

Métodos	# de Atributos	Outex	USPTex	Brodatz	Vistex
GLCM	24	80,73	83,64	90,43	92,24
GLDM	60	86,76	92,06	94,43	97,11
Filtros de Gabor	48	81,91	89,22	89,86	93,29
Fourier	63	81,91	67,50	75,90	79,51
Fractal	69	80,51	78,27	87,16	91,67
Fractal Fourier	68	68,38	59,47	71,96	79,75
LBP	256	81,10	85,43	93,64	97,92
LBPV	555	75,66	54,97	86,26	88,65
CLBP	648	85,80	91,14	95,32	98,03
AHP	120	88,31	94,85	94,88	98,38
BSIF	256	77,43	77,66	91,44	88,66
LCP	81	86,25	91,14	93,47	94,44
LFD	276	82,57	83,55	90,99	94,68
LPQ	256	79,41	85,12	92,51	92,48
ELM	180	89,71	95,11	95,27	98,15
CNTD	108	86,76	91,71	95,27	98,03
VGG19	512	76,62	93,19	96,79	97,45
InceptionV3	2048	86,40	96,77	98,54	98,84
ResNet50	2048	65,66	62,30	81,98	81,71
InceptionResNetV2	1536	85,88	96,34	98,99	98,96
Métodos Propostos					
ERCNN ¹	90	89,34	95,46	95,05	98,61
ERCNN ²	180	91,54	96,64	96,11	98,73
ERCNN ³	240	91,32	96,95	96,06	99,19
PLCNN	330	92,13	97,21	98,09	99,88

Fonte: Elaborada pelo autor.

de dados de texturas rotacionadas foram usadas: Brodatz, Outex e Vistex. Nessas bases de dados, cada imagem de textura original é rotacionada em um ângulo específico, totalizando 11 ângulos diferentes, conforme o procedimento adotado por [Backes, Casanova e Bruno \(2013\)](#). Assim, cada classe de textura tem como amostras os 11 ângulos diferentes. Por exemplo, a base Brodatz tem 111 imagens originais de textura (cada imagem é uma classe), totalizando $11 \times 111 = 1221$ amostras. A Vistex e a Outex têm 594 (54 classes de textura) e 748 (68 classes de textura) amostras, respectivamente. A [Figura 23](#) mostra exemplos de uma imagem de textura rotacionada em diferentes ângulos. Para validação, um esquema *leave-one-out* foi usado neste experimento. Nesse esquema, uma textura rotacionada em um determinado ângulo é utilizada para teste, enquanto as amostras restantes representando os outros 10 ângulos, bem como as amostras das outras classes, são usadas para treinar o classificador LDA. Em outras palavras, como é usada uma única imagem em diferentes rotações para cada classe, este experimento mede a taxa de tolerância à rotação dos métodos.

As acurácias obtidas para os métodos propostos e outros métodos da literatura nas três bases de dados rotacionadas são mostradas na [Tabela 6](#). Esses resultados demonstram que os métodos são robustos à rotação, uma vez que apresentam acurácias altas na base de dados. Em

Figura 23 – Exemplos de uma textura da base Brodatz (BRODATZ, 1966) rotacionadas em 11 diferentes ângulos. Da esquerda para a direita, de cima para baixo: 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°.

Fonte: Ribas *et al.* (2020), Brodatz (1966).

particular, o método ERCNN obteve a melhor acurácia na Vistex (empatado com as assinaturas AHP, LCP e ELM), a segunda melhor acurácia na base de dados Brodatz (junto com as assinaturas ELM e AHP) e a terceira melhor taxa de sucesso na Outex. Também é importante ressaltar a diferença muito pequena no desempenho da representação ERCNN³ quando comparado às melhores acurácias atingidas na Outex (0,66%) e Brodatz (0,24%). No caso do método PLCNN, o desempenho foi inferior, principalmente, na base de dados Outex. Isso ocorre devido à forma de construir o vetor de entrada para cada pixel/vértice, que usa uma ordem fixa para percorrer os oito vizinhos.

Por outro lado, o método com menor robustez à rotação foi o BSIF. Observe também que alguns métodos que tiveram um bom desempenho nas bases de dados originais (Tabela 5) não funcionam muito bem com imagens rotacionadas. Por exemplo, na Tabela 5, o descritor LBP obteve as maiores acurácias quando comparado aos métodos LPQ e LBPV, no entanto, seus resultados são inferiores às acurácias desses dois métodos no experimento de rotação. Isso reforça o quão discriminativo são as abordagens desenvolvidas, visto que obtiveram bom desempenho tanto nas bases de dados rotacionadas quanto nas não rotacionadas.

Análise de Tempo de Processamento

Para melhorar a avaliação dos métodos, o tempo médio de execução das abordagens para cada imagem processada é mostrado na Tabela 7. Nos experimentos, um processador 3.60GHz Intel(R) Core i7-7820X, 64GB de memória RAM e sistema operacional de 64 bits foram utilizados. O método LBP possui o menor tempo computacional, enquanto a abordagem de assinatura baseada em ELM obteve o maior tempo de processamento, que pode ser explicado pelo esquema de combinação linear usado para robustez à rotação. Por outro lado, os resultados dos métodos propostos indicam um tempo de execução competitivo para aplicações de tempo real em relação aos demais métodos. A abordagem proposta ERCNN levou, em média, 0,1132s, 0,2538s e 0,4619s nas versões ERCNN¹, ERCNN² e ERCNN³, respectivamente. Por outro lado, a abordagem PLCNN obteve 0,2884s para o cálculo da assinatura. Note que a versão proposta

Tabela 6 – Comparação de acurácias de diferentes métodos de análise de textura em bases de dados rotacionadas. Um subconjunto dos resultados comparados está presente em (JUNIOR; BACKES, 2016).

Métodos	# de Atributos	Outex	Brodatz	Vistex
GLCM	24	75,53	91,48	94,28
GLDM	60	84,76	95,17	99,16
Filtros de Gabor	48	78,07	90,17	96,97
Fourier	63	88,90	94,59	97,64
Fractal	69	86,76	95,99	97,47
Fractal Fourier	68	81,02	89,68	97,31
LBP	256	71,39	91,97	94,44
LBPV	555	89,83	98,11	98,48
CLBP	648	97,59	99,18	99,49
AHP	120	92,25	99,43	99,83
BSIF	256	65,78	82,88	85,19
LCP	81	93,58	98,53	99,83
LFD	276	92,51	98,53	99,33
LPQ	256	90,64	99,67	98,99
ELM	180	97,99	99,43	99,83
CNTD	108	85,56	94,66	98,48
Métodos Propostos				
ERCNN ¹	90	95,86	99,10	99,83
ERCNN ²	180	96,52	99,43	99,83
ERCNN ³	240	97,33	99,43	99,83
PLCNN	330	91,84	98,36	99,66

Fonte: Elaborada pelo autor.

mais rápida (ERCNN¹) utiliza somente o raio de conexão $R = 4$, enquanto as demais utilizam 2 raios combinados, o que leva mais tempo para processar.

Discussões e Considerações Finais

Como mostrado nos resultados acima, os dois métodos propostos obtiveram desempenhos promissores combinados com aspectos interessantes para a área e problemas de análise de texturas. Os resultados também confirmaram que a combinação de uma abordagem baseada em aprendizado de representação por RNNs com uma modelagem de redes complexas direcionadas possui grande potencial. Isso também é evidenciado na considerável melhora dos resultados quando comparado os métodos propostos com os descritores ELM e CNDT, que são baseados em redes neurais randomizadas e redes complexas, respectivamente. Enquanto o ELM usa somente informações dos pixels para treinar as RNNs, a abordagem CNDT utiliza uma modelagem de redes complexas (diferente da usada aqui) e medidas estatísticas para caracterizar a rede.

Em termos de tamanho do vetor de características, embora os métodos propostos possuam representações com maior número de atributos em comparação à alguns métodos da literatura, é importante ressaltar que, se considerar apenas a versão ERCNN¹, os resultados ainda são

Tabela 7 – Comparação do tempo de processamento dos métodos de análise de texturas em níveis de cinza para extração de assinatura de uma imagem.

Métodos	Tempo (s)
GLCM	0,0167
GLDM	0,0051
Filtros de Gabor	0,0392
Fourier	0,0148
Fractal	0,5092
Fractal Fourier	0,0038
LBP	0,0032
LBPV	0,2005
CLBP	0,4740
AHP	0,0649
BSIF	0,0051
LCP	0,0398
LFD	0,1067
LPQ	0,1376
ELM	1,0067
CNTD	0,0851
Métodos Propostos	
ERCNN ¹	0,1132
ERCNN ²	0,2538
ERCNN ³	0,4619
PLCNN	0,2884

Fonte: Elaborada pelo autor.

competitivos. Nesse caso, a assinatura possui apenas 90 atributos, leva um tempo baixo para processar as imagens e fornece acurácias competitivas em várias bases. Já o método PLCNN alcançou maiores acurácias na comparação tanto com o método ERCNN, bem como em relação aos demais métodos da literatura, além de menor tempo de processamento quando comparado a versão ERCNN³. Esses resultados indicam que a melhor forma de analisar as redes complexas modeladas é através dos padrões locais (vizinhança de 8) dos pixels/vértices usados pelo PLCNN, enquanto o ERCNN aprende as características locais indiretamente através da evolução da topologia das redes complexas. De fato, caracterizar padrões locais é um passo chave em análise de texturas, apesar de muitas vezes a análise multi-escala também ser necessária.

Por outro lado, a estratégia de análise de padrões locais complexos também interfere na capacidade do método PLCNN em lidar com imagens rotacionadas, fazendo com o seu desempenho seja inferior ao ERCNN. Isso é devido à estratégia de usar uma ordem fixa para percorrer a vizinhança de 8 e obter o vetor de atributos de cada vértice. Embora não afete drasticamente na tolerância à rotação do método, investigações futuras podem ser realizadas para superar essa desvantagem. A utilização de uma combinação linear para definir a ordem dos vizinhos é um exemplo de estratégia futura para solucionar essa desvantagem, como adotado por [Junior e Backes \(2016\)](#), apesar de ser importante destacar que isso representa um aumento no

tempo computacional de processamento.

Por fim, algumas considerações importantes podem ser feitas. As taxas de acurácia das abordagens desenvolvidas foram altas, ultrapassando a acurácia de um grande conjunto de métodos comparados, incluindo alguns baseados em aprendizado profundo. Os métodos desenvolvidos também alcançaram um bom desempenho nos diversos aspectos considerados, tais como, o número de atributos gerados, robustez a rotação e tempo de processamento computacional. Esse equilíbrio faz com que as abordagens propostas e suas diferentes versões se apresentem como uma boa opção para vários problemas e aplicações que envolvem texturas em níveis de cinza. Assim, acredita-se que esses estudos ofereçam contribuições relevantes para esse novo e promissor campo de pesquisa que investiga como conectar redes neurais e complexas para construir representações de imagens.

5.2 Aprendizado de Representação Espaço-Espectral para Análise de Texturas Coloridas (SSRNN)

Na literatura, existem muitas técnicas de análise de texturas desenvolvidas para imagens em níveis de cinza. No entanto, hoje em dia, as imagens coloridas são capturadas pela maioria dos dispositivos de imagem e estão presentes em muitos conjuntos de dados e aplicações. Assim, diferentes abordagens para análise de textura têm sido propostas visando caracterizar a textura em imagens coloridas (MÄENPÄÄ; PIETIKÄINEN, 2004; CERNADAS *et al.*, 2017; SCABINI *et al.*, 2019a). A maioria delas são extensões que empregam as técnicas de textura em escala de cinza nos canais de imagem separadamente (métodos integrativos) ou transformam a imagem colorida em escala de cinza. Por outro lado, métodos de cor pura são abordagens que consideram apenas as cores da imagem, sem considerar a relação espacial do pixel. Portanto, uma abordagem comum é combinar esses métodos e características de nível de cinza em métodos paralelos. Além disso, recentemente, modelos de redes neurais convolucionais profundas têm sido empregados na caracterização de imagens coloridas para diferentes tarefas (SZEGEDY *et al.*, 2016; CIMPOI *et al.*, 2016). No entanto, muitas dessas abordagens não lidam com as informações de cores inteiramente, como padrões espaciais entre os pixels multi-espectrais.

Esta seção propõe uma abordagem para analisar imagens de textura colorida, que é baseado em uma representação espaço-espectral (em inglês, *Spatio-Spectral Representation* (SSR)) aprendida por RNNs a partir do modelo de redes complexas direcionada SSN (descrita na Seção 4.2). Assim, esta abordagem é nomeada como *Spatio-Spectral Representation learned by RNN* (SSRNN). As imagens de textura colorida são modeladas como SSNs direcionadas, que definem os pixels para todos os canais como vértices e criam conexões entre os mesmos e diferentes canais empregando uma vizinhança radialmente simétrica. Essa modelagem foi publicada no trabalho da referência: (SCABINI; RIBAS; BRUNO, 2020). A SSN resultante fornece informações espaciais por meio da vizinhança simétrica e as conexões entre canais

representam as características espectrais. A RNN é treinada a partir das métricas dos vértices (grau e força) para prever a dinâmica de evolução da SSN à medida que o raio r é variado, e os pesos aprendidos da camada de saída são usados como um vetor de características, chamado de representação espaço-espectral. Assim, a contribuição é uma caracterização discriminativa da SSN a partir da representação aprendida pela RNN, que realiza uma descrição das propriedades espaciais e espectrais da imagem colorida. A seguir a abordagem proposta, os resultados e discussões em diferentes bases de imagens são apresentadas.

5.2.1 Representação Espaço-Espectral

A representação espaço-espectral proposta é formada por características de textura aprendidas por RNNs usando as medidas de centralidade calculadas da SSN modelada, que são o grau de saída, a força de saída e a força de entrada. Essas são medidas simples que podem ser calculadas rapidamente na etapa de modelagem da rede complexa, visto que usa apenas os vértices vizinhos. Apenas o grau de saída $\bar{k}$ foi utilizado como entrada, pois o grau de entrada $\hat{k}$ e saída são uma transformação linear em função do raio máximo, ou seja, $\forall v_i \in V, \bar{k}_{v_i} + \hat{k}_{v_i} = k_{max}^r$, em que k_{max}^r é o número máximo de conexões para um raio r . Por outro lado, como a força do vértice é calculada com base na distribuição das intensidades dos pixels, não há correlação entre a força de saída e entrada. Portanto, as medidas utilizadas para a descrição do vértice são: grau de saída $\bar{k}_{v_i}$, força de saída $\bar{s}_{v_i}$ e força de entrada $\hat{s}_{v_i}$.

As medidas de redes complexas das SSNs modeladas contêm informações importantes sobre a textura que representam. A [Figura 24](#) mostra os resultados da reconstrução da imagem RGB original de cada medida (grau de saída $\bar{k}$, força de entrada $\hat{s}$ e força de saída $\bar{s}$) para as RCs: G , G_S e G_T . É possível perceber como cada medida e rede complexa destacam diferentes propriedades visuais da textura de entrada. Quando combinadas, eles formam uma representação enriquecida que a RNN pode aprender para caracterização da textura.

Para tirar proveito dos diferentes padrões de textura capturados por cada medida de vértice, matrizes de entrada para treinar as RNNs foram criadas usando essas três medidas. Essas matrizes são construídas usando uma estratégia de análise da dinâmica de evolução da SSN para diferentes valores do parâmetro de modelagem r , conforme ilustrado na [Figura 25](#). Portanto, para um vértice v_i , os graus de saída para diferentes valores de raio $r = \{1, 2, \dots, R-1\}$ formam o vetor de entrada $\vec{x}_{v_i} = [\bar{k}_{v_i}^1, \bar{k}_{v_i}^2, \dots, \bar{k}_{v_i}^{R-1}]$ que será usado para prever o grau de saída de v_i para o raio máximo R , sendo o seu rótulo $y_i = \bar{k}_{v_i}^R$. Esse processo é realizado para todos os vértices da SSN na versão original (G) para obter a matriz de atributos de entrada $X_{\bar{k}}^G$ e o vetor de rótulo $Y_{\bar{k}}^G$ representando o grau de saída (veja a [Figura 25](#) (b)). Assim, pode-se analisar a evolução topológica dos vértices a medida que a sua vizinhança é incrementada, atingindo regiões maiores e realizando uma análise multi-escala. Da mesma forma, matrizes de atributos de entrada foram construídas usando a força de saída $X_{\bar{s}}^G$ e a força de entrada $X_{\hat{s}}^G$. Para os casos da força de saída $X_{\bar{s}}^G$ e a força de entrada $X_{\hat{s}}^G$ os vetores de rótulos são $Y_{\bar{s}}^G$ e $Y_{\hat{s}}^G$, respectivamente. Além disso, as

Figura 24 – Representação visual da SSN ($R = 8$); imagens resultantes são obtidas pelo cálculo de medidas de RC para um vértice em cada camada da rede, para depois ser normalizada em pixels RGB de 8-bits. G_S , G e G_T referem-se aos grafos espacial, completo e espectral, respectivamente.

Fonte: Elaborada pelo autor.

matrizes de entrada também foram construídas usando medidas calculadas a partir das redes complexas formadas por conexões dentro do canal G_S e as redes com conexões entre canais G_T , resultando nas seguintes matrizes de entrada: $X_{\bar{k}}^{G_S}$, $X_{\bar{s}}^{G_S}$, $X_{\hat{s}}^{G_S}$, $X_{\bar{k}}^{G_T}$, $X_{\bar{s}}^{G_T}$, $X_{\hat{s}}^{G_T}$.

O próximo passo para treinar a rede neural é definir a matriz de peso da camada de entrada. Nesse caso, a mesma estratégia da [Seção 5.1](#) foi adotada, ou seja, o gerador pseudo-aleatório LCG utilizando sementes fixas é empregado, similarmente às abordagens anteriores. A matriz de peso H e as matrizes de entrada X são normalizadas com z -score. Nesta proposta, os vetores de características que formam as representações espaço-espectrais para analisar a imagem de textura colorida são criados usando a matriz de pesos treinados, que é definida como o vetor $\vec{f} = YZ^T(ZZ^T + \lambda I)^{-1}$, em que $\lambda = 10^{-3}$ e seu comprimento é $Q + 1$ devido ao valor de bias ([Figura 26](#) (c)-(d)). Assim, nove RNNs são treinadas com as seguintes matrizes de atributos de entrada $X_{\bar{k}}^G$, $X_{\bar{s}}^G$, $X_{\hat{s}}^G$, $X_{\bar{k}}^{G_S}$, $X_{\bar{s}}^{G_S}$, $X_{\hat{s}}^{G_S}$, $X_{\bar{k}}^{G_T}$, $X_{\bar{s}}^{G_T}$ e $X_{\hat{s}}^{G_T}$, produzindo os vetores $\vec{f}$ que são combinados para obter a primeira representação espaço-espectral,

Figura 25 – Ilustração da criação da matriz de atributos de entrada X_k^G com o grau de saída da versão completa G . A imagem da textura colorida é modelada em redes complexas com diferentes valores de r (a figura mostra um exemplo para $r = 1$). O grau de saída $\bar{k}_{v_i}$ dos vértices para diferentes valores de r são usados para criar a matriz de entrada X_k^G e seu grau de saída $\bar{k}_{v_i}^R$ para raio R é usado como rótulo.

Fonte: Elaborada pelo autor.

$$\vec{Y}(R)_Q = \left[\vec{f}_{\bar{k}}^G, \vec{f}_{\bar{s}}^G, \vec{f}_{\hat{s}}^G, \vec{f}_{\bar{k}}^{G_S}, \vec{f}_{\bar{s}}^{G_S}, \vec{f}_{\hat{s}}^{G_S}, \vec{f}_{\bar{k}}^{G_T}, \vec{f}_{\bar{s}}^{G_T}, \vec{f}_{\hat{s}}^{G_T} \right], \quad (5.11)$$

em que R é o número de raios usados. O vetor $\vec{Y}(R)_Q$ é produzido com um único valor de Q e R . Esses dois parâmetros influenciam no processo de aprendizado da rede neural e, consequentemente, nas características aprendidas. Nesse contexto, vetor $\vec{\Theta}(Q)_{(R_1, R_2)}$ que combina os vetores $\vec{Y}(R)_Q$ para dois valores de R é proposto,

$$\vec{\Theta}(Q)_{(R_1, R_2)} = \left[\vec{Y}(R_1)_Q, \vec{Y}(R_2)_Q \right]. \quad (5.12)$$

Da mesma forma, também é proposta a representação final $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ que combina os vetores $\vec{\Theta}(Q)_{(R_1, R_2)}$ com diferentes valores de Q ,

$$\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m} = \left[\vec{\Theta}(Q_1)_{(R_1, R_2)}, \vec{\Theta}(Q_2)_{(R_1, R_2)}, \dots, \vec{\Theta}(Q_m)_{(R_1, R_2)} \right]. \quad (5.13)$$

A Figura 26 apresenta o processo geral da metodologia proposta para uma imagem colorida e usando um único valor de parâmetros Q e R . Note que nesse exemplo são consideradas as três matrizes de entrada referentes aos tipos de SSNs: G , G_T e G_S . Por outro lado, em relação à medida que compõe as matrizes, pode-se utilizar uma das três medidas de vértices consideradas ($\bar{k}$, $\bar{s}$ e $\hat{s}$), como já descrito anteriormente.

Figura 26 – Esquema geral do método proposto SSRNN para uma imagem colorida e valores de parâmetros R e Q . Primeiramente, a imagem colorida é representada pela modelagem SSN direcionada, resultando em três RCs: G , G_S e G_T . Observe que nessa etapa o raio de conexão varia de $\{1, 2, \dots, R-1\}$. Em seguida, as matrizes de atributos de entrada X (usando uma das medidas $\bar{k}$, $\bar{s}$ e $\hat{s}$) são criadas para as três RCs com base no conjunto de raios e então as RNNs são treinadas para prever o conjunto Y formado pelos graus ($\bar{k}$ ou $\hat{k}$) para o raio R . Por fim, os vetores da camada de saída com os pesos aprendidos compõem o vetor de características.

Fonte: Elaborada pelo autor.

5.2.2 Resultados e Discussões

Esta seção apresenta os experimentos, resultados e comparações para o método proposto e demais métodos da literatura em três bases de dados de textura colorida.

Configuração Experimental

Nos experimentos, o classificador LDA é empregado para avaliar e comparar o desempenho do método proposto e dos demais. Na análise dos parâmetros, o esquema de validação cruzada de 10 pastas é adotado. O esquema de validação cruzada *leave-one-out* também foi usado na análise comparativa. Em todos os experimentos, a acurácia de classificação é usada como medida de desempenho (ou seja, a porcentagem de imagens classificadas corretamente). Para avaliar os métodos, três bases de dados com texturas coloridas foram utilizadas: Outex, USPTex e MBT. As bases de dados são descritas com detalhes no [Apêndice A](#).

Estudo de Parâmetros

Para analisar os valores dos parâmetros, o método proposto foi empregado nas três bases de dados de textura colorida usando diferentes valores dos parâmetros Q e R . As primeiras representações analisadas são $\vec{Y}(R)_Q$ e $\vec{\Theta}(Q)_{(R_1, R_2)}$ para diferentes valores de $R = \{04, 06, 08, 10, 12\}$ e suas combinações. Para calcular essa representação, foi considerado um valor fixo $Q = 4$.

O parâmetro R determina o número de raios usados na análise da dinâmica de evolução da rede complexa e também determina as matrizes de atributos de entrada X usadas para treinar a rede neural. A Figura 27 mostra a matriz com as acurácias médias calculadas nas três bases de dados para um valor de R (diagonal principal) e dois valores combinados. Observe que as maiores acurácias são alcançadas quando é combinado dois valores de R . Esse comportamento é esperado: mais valores de raio combinados, maior o tamanho do vetor de atributos. Nesse caso, a maior acurácia é atingida com $\vec{\Theta}(04)_{(04,10)}$. Isto é, as redes complexas serão analisadas com dois conjuntos de raios: $r = \{1, 2, 3, 4\}$ e $r = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Os raios menores realizam uma análise local da vizinhança do pixel, por sua vez, à medida que o raio é incrementado, regiões mais distantes são acessadas e consideradas na topologia da SSN, permitindo uma análise multi-escala da imagem.

Figura 27 – Acurácia média nas três bases de dados para diferentes valores de R e suas combinações com $Q = 04$.

Fonte: Elaborada pelo autor.

Um segundo experimento avalia a representação $\vec{\Psi}(R_1, R_2)_{Q_1, Q_2, \dots, Q_m}$ usando diferentes valores e combinações de Q . Nesse experimento, são testados os seguintes valores de $Q = \{04, 09, 14, 19, 24\}$ com os melhores valores de raio $R = \{04, 10\}$, que foi encontrado anteriormente. A Figura 28 mostra as matrizes de acurácias calculadas para as três bases de dados. Nesse teste, os resultados indicam que dois valores combinados de Q são melhores do que um valor único (diagonal principal) em todas as bases de dados. No entanto, os resultados usando um valor único de Q são competitivos porque produzem vetores de atributos menores e também menor tempo de processamento computacional.

O desempenho da combinação de três valores de Q nas três bases de dados também foi avaliado, como pode ser observado na Tabela 8. Para esse caso, os resultados indicam desempenhos semelhantes nos vários valores combinados de Q que foram testados. Quando considerada a compensação entre acurácia e dimensão do vetor de atributos em todas as bases

de dados, a melhor combinação é $Q = \{04, 09, 19\}$. Desta forma, as acurácias alcançadas foram $97,13 (\pm 0,88)$, $98,17 (\pm 0,70)$ e $99,43 (\pm 0,36)$ na Outex, MBT, USPTex, respectivamente. Além dessa configuração, também pode ser considerada a $Q = \{09, 19, 24\}$, pois produz acurácias altas nas bases de dados, embora o seu vetor de características seja maior. Portanto, os seguintes vetores de características serão considerados como representações espaço-espectral padrão do método: $\vec{\Psi}(04, 10)_{04,09,19}$ (SSRNN¹) e $\vec{\Psi}(04, 10)_{09,19,24}$ (SSRNN²).

Figura 28 – Acurácias para o método SSRNN para diferentes valores de Q (diagonal principal) e suas combinações.

Fonte: Elaborada pelo autor.

Comparações e Discussões

Para enriquecer a análise do método proposto, o seu desempenho foi comparado com os resultados obtidos por outros métodos da literatura. Para tanto, os resultados do método proposto foram calculados com as duas representações espaço-espectral (SSRNN¹ e SSRNN²) que foram escolhidas na análise anterior. Para o método proposto SSRNN, o classificador LDA foi adotado

Tabela 8 – Resultados para o vetor de características $\tilde{\Psi}(04, 10)_{Q_1, Q_2, \dots, Q_m}$ combinando $\{Q_1, Q_2, Q_3\}$ para os raios $R = \{4, 10\}$.

$\{Q_1, Q_2, Q_3\}$	Outex	MBT	USPTex
{04, 09, 14}	96,40 ($\pm 1,49$)	98,38 ($\pm 0,77$)	99,43 ($\pm 0,36$)
{04, 09, 19}	97,13 ($\pm 0,88$)	98,17 ($\pm 0,70$)	99,43 ($\pm 0,36$)
{04, 09, 24}	96,76 ($\pm 1,31$)	98,17 ($\pm 0,64$)	99,26 ($\pm 0,80$)
{04, 14, 19}	96,54 ($\pm 0,85$)	97,85 ($\pm 0,66$)	99,52 ($\pm 0,38$)
{04, 14, 24}	96,62 ($\pm 1,44$)	97,93 ($\pm 0,56$)	99,39 ($\pm 0,51$)
{04, 19, 24}	96,47 ($\pm 1,19$)	98,21 ($\pm 0,77$)	99,17 ($\pm 0,66$)
{09, 14, 19}	96,76 ($\pm 1,05$)	97,97 ($\pm 0,51$)	99,39 ($\pm 0,37$)
{09, 14, 24}	96,54 ($\pm 0,98$)	98,25 ($\pm 0,81$)	99,26 ($\pm 0,36$)
{09, 19, 24}	97,13 ($\pm 1,27$)	98,21 ($\pm 0,64$)	99,21 ($\pm 0,61$)
{14, 19, 24}	97,21 ($\pm 1,33$)	96,96 ($\pm 1,13$)	99,13 ($\pm 0,41$)

Fonte: Elaborada pelo autor.

usando dois esquemas de validação cruzada: *leave-one-out* (loo) e 10 pastas. Para a comparação foram considerados os métodos da literatura tradicionais ou "artesanais" e também os baseados em aprendizado de características. Os métodos tradicionais são: Opponent-Gabor (264) (JAIN; HEALEY, 1998), *Complete Local Binary Patterns* (CLBP) (354) (GUO; ZHANG; ZHANG, 2010a), *Local Phase Quantization* (LPQ) (768) (OJANSIVU; HEIKKILÄ, 2008), *Complex Networks Traditional descriptors* (CNTD) (324) (BACKES; CASANOVA; BRUNO, 2013), *Multilayer Complex Network Descriptors* (MCND) (SCABINI *et al.*, 2019a), SSN (ϕ_{WB}^6) (648) (SCABINI; RIBAS; BRUNO, 2020) e RNN-RGB (180) (JUNIOR; BACKES; BRUNO, 2019). Os valores em parênteses representam o número de atributos gerados pelos métodos.

A Tabela 9 resume os resultados obtidos para cada método, incluindo o método proposto SSRNN. Observe que o método proposto obtém uma maior acurácia na base de dados Outex quando comparado com os demais métodos tradicionais. Na base de dados USPTex, o método proposto obteve acurácia superior a todos os outros, exceto em relação ao método SSN em que a acurácia foi levemente inferior (0,4%). Para a base MBT, o desempenho do método também foi competitivo, alcançando uma acurácia próxima aos métodos SSN, CNDT e CLBP.

Em relação aos métodos baseados em aprendizado, os resultados do método desenvolvido foram comparados com as seguintes arquiteturas de rede neural convolucional: AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), VGG16 e VGG19 (SIMONYAN; ZISSERMAN, 2014), ResNet50 e ResNet101 (HE *et al.*, 2016), e InceptionV3 (SZEGEDY *et al.*, 2015; SZEGEDY *et al.*, 2016). Os resultados foram obtidos usando uma abordagem de extração de características com as camadas pré-treinadas na base de dados ImageNet (DENG *et al.*, 2009), que estão disponíveis no programa Matlab. As imagens são redimensionadas conforme o tamanho da camada de entrada das CNNs e as camadas convolucionais com tamanho menor do que 800 são usadas para calcular as características da imagem. Em seguida, o GAP é aplicado nos mapas de ativação gerados para obter o vetor de características, como usado previamente

em outros trabalhos (LIN; CHEN; YAN, 2013; CIMPOI *et al.*, 2014; CIMPOI *et al.*, 2016). Assim, o tamanho dos vetores de atributos calculados são de 768, 512, 256 e 512 descritores para os modelos InceptionV3, ResNet, AlexNet e VGG, respectivamente. A descrição e resultados experimentais com esses modelos também estão no artigo do SSN (SCABINI; RIBAS; BRUNO, 2020). Quando comparado às arquiteturas de CNN, o método proposto obteve maior acurácia nos conjuntos de dados Outex e MBT. Por outro lado, a ResNet50 alcança acurácia levemente superior à abordagem SSRNN na USPTex. Nas bases de dados Outex e MBT, o método proposto melhora, significativamente, a acurácia em comparação com várias arquiteturas de CNNs, por exemplo, de 91,5% (ResNet50) para 96,8 % na base Outex e de 94,9% (ResNet50) para 98,20% na MBT. Em particular, na base de dados Outex, que é a mais desafiadora por ser composta por amostras com diferentes iluminações, pode-se observar que todos os métodos baseados em CNN não alcançam um bom desempenho. Dessa forma, esses resultados indicam que o método SSRNN pode atingir resultados competitivos usando uma rede neural simples e rápida, enquanto as CNNs são maiores e computacionalmente caras.

No que diz respeito aos métodos baseados na teoria de redes complexas e grafos, a abordagem proposta foi comparada com três métodos diferentes: CNDT em nível de cinza e integrativo, MCND e SSN. Nessa comparação, o método SSRNN superou os outros na base Outex, enquanto o SSN obteve acurácias levemente superiores nas bases USPTex e MBT. Os dois métodos também geram vetores de características de tamanhos similares, isto é, 648 atributos para o SSN e 630 atributos no caso do método SSRNN¹. Isso demonstra que a proposta de usar uma rede neural simples para aprender e descrever a topologia da SSN pode melhorar a análise de texturas coloridas em muitos casos em comparação às outras medidas tradicionais de rede complexa e teoria de grafos. Já quando o método SSRNN é comparado com a versão do RNN aplicada diretamente na imagem RGB, o método proposto melhora as acurácias nas duas bases comparadas (USPTex e Outex), evidenciando o poder da combinação de redes complexas e RNNs para reconhecimento de padrões.

Por fim, em contraste com métodos que possuem limitações de desempenho em bases de dados específicas, o método proposto com suas duas representações desempenha um resultado médio consistente ao considerar todos os cenários. Por exemplo, na base de dados USPTex, as abordagens baseadas em CNN superam o método proposto em até 0,4%, enquanto o método proposto aumenta consideravelmente a acurácia na Outex e também na MBT em comparação com estas abordagens. Do mesmo modo, os descritores integrativos também mostram resultados variáveis conforme as bases de dados, destacando as acurácias bem inferiores na base de dados Outex.

5.2.3 Considerações Finais

Esta seção introduz um novo método para caracterizar texturas coloridas que aprende representações espaço-espectral a partir da modelagem "artesanal" de redes complexas SSN.

Tabela 9 – Resultados de comparação para o método proposto e métodos da literatura usando o classificador LDA (*leave-one-out*) em três bases de dados de texturas coloridas. Resultados do método proposto com * foram calculados com validação cruzada em 10 pastas. As acurácias do método RNN-RGB foram extraídas diretamente do artigo original, em que o resultado para a base MBT não está disponível.

Métodos	USPTex	Outex	MBT
Opponent-Gabor	99,1	93,5	97,6
LPQ integrativo	90,4	80,1	95,7
CLBP integrativo	97,4	89,6	98,2
CNTD (nível de cinza)	92,3	86,8	83,7
CNTD integrativo	97,9	92,3	98,5
RNN-RGB	98,4	94,8	-
MCND	99,0	95,4	97,0
SSN	99,7	96,6	98,6
AlexNet	99,6	91,4	97,8
VGG16	99,5	91,1	97,2
VGG19	99,5	90,7	96,3
InceptionV3	99,5	89,5	94,4
ResNet50	99,7	91,5	94,9
ResNet101	99,5	91,3	94,6
Métodos Propostos			
SSRNN ^{1*}	99,43 ($\pm 0,36$)	97,13 ($\pm 0,88$)	98,17 ($\pm 0,70$)
SSRNN ^{2*}	99,21 ($\pm 0,61$)	97,13 ($\pm 1,27$)	98,21 ($\pm 0,64$)
SSRNN ¹	99,3	96,7	98,2
SSRNN ²	99,0	96,8	98,0

Fonte: Elaborada pelo autor.

Dois vetores de características que são as representações espaço-espectrais finais do método foram propostos: SSRNN¹ ($\vec{\Psi}(04,10)_{04,09,19}$) e SSRNN² ($\vec{\Psi}(04,10)_{09,19,24}$). Os resultados superaram quase todos os métodos comparados nas três bases de dados, demonstrando que a abordagem proposta produz representações espaço-espectral altamente discriminativas. O método proposto também provou ser muito competitivo em comparação com as arquiteturas de CNN profundas. Além disso, os resultados promissores reforçam mais uma vez o grande potencial na combinação de uma modelagem artesanal de textura (SSN) e uma representação aprendida usando a RNN como metodologia de análise de imagens. O potencial da abordagem proposta também é demonstrado na aplicação de caracterização da plasticidade foliar de espécies de Jacaranda Caroba como um biomarcador natural de poluentes, como pode ser visto na [Seção 8.2](#). Como trabalhos futuros, acredita-se que duas estratégias podem ser exploradas para melhorar os resultados: (i) usar uma RNN mais profunda para aprender as características; (ii) empregar um modelo de classificação de RNN em vez de um modelo de regressão para treinar a rede neural.

5.3 Redes Complexas e RNNs para Análise de Texturas Dinâmicas

Texturas dinâmicas (do inglês, *Dynamic Texture* (DT)) são padrões visuais que variam tanto espacial quanto temporalmente. Portanto, estas podem representar uma ampla gama de fenômenos naturais que variam no tempo, tais como, fogo, ondas do mar, fontes de água, fumaça, multidões, entre outros. As dinâmicas das texturas estão relacionadas aos objetos ou processos presentes nos vídeos, que apresentam uma certa estacionariedade em relação ao espaço e ao tempo. (DORETTO *et al.*, 2003). Em geral, estas podem ser caracterizadas por padrões simples e repetitivos ou movimento não rígido guiado por dinâmicas não lineares e estocásticas, como ramos balançando ou água borbulhante (XU *et al.*, 2011; RIBAS; GONÇALVES; BRUNO, 2019). Nos últimos anos, as DTs têm recebido atenção significativa da comunidade de visão computacional devido às suas diversas aplicações, como detecção de face (Zhao; Lin; Heikkilä, 2018), monitoramento de tráfego (DERPANIS; WILDES, 2011), análise de comportamento de multidões (Li; Mahadevan; Vasconcelos, 2014), detecção de incêndio (Dimitropoulos; Barmoutis; Grammalidis, 2015) etc.

A tarefa básica de reconhecimento de DT consiste em classificar um trecho de vídeo (ou seja, um volume de dados 3D espaço $\times$ tempo), com relação a classes significativas, como as mencionadas anteriormente. Essa tarefa pode então ser estendida a outros objetivos, como segmentação semântica de superfícies dinâmicas ou detecção discriminativa de objetos em fundos não estacionários. A notória dificuldade de reconhecimento de texturas dinâmicas vem da grande variabilidade e do caráter difuso das texturas naturais que, ao contrário dos objetos, não podem - em geral - ser precisamente localizadas no espaço e no tempo. A esses desafios são adicionados aqueles específicos da natureza dinâmica do problema: o modelo deve representar os diferentes tipos de movimento que podem afetar a textura: fluxo, ondulação, crescimento etc. Além disso, como um passo básico, espera-se que o reconhecimento de DT funcione de forma rápida e reativa. Isto é, ele deve ser capaz de funcionar em tempo real e se adaptar on-line a um ambiente em mudança.

Motivado por esses desafios, uma série de abordagens que podem ser divididas em cinco categorias foram propostas: (i) baseados em filtros (WILDES; BERGEN, 2000; DERPANIS; WILDES, 2010); (ii) baseados em movimentos (POLANA; NELSON, 1997; RIVERA; CHAE, 2015); (iii) baseados em discriminação (ZHAO; PIETIKÄINEN, 2006; TIWARI; TYAGI, 2017; TIWARI; TYAGI, 2016a); (iv) baseados em aprendizagem (QI *et al.*, 2016; ANDREARCZYK; WHELAN, 2018); e (v) métodos baseados em modelos (RAVICHANDRAN; CHAUDHRY; VIDAL, 2009; GONÇALVES; MACHADO; BRUNO, 2015; RIBAS; GONÇALVES; BRUNO, 2019).

Em particular, métodos baseados em redes complexas obtiveram ótimos resultados devido à sua flexibilidade e capacidade de representar o movimento e a aparência nas DTs (RIBAS;

GONÇALVES; BRUNO, 2019; GONÇALVES; MACHADO; BRUNO, 2015; RIBAS; BRUNO, 2020). Esses métodos modelam o vídeo como redes complexas e extraem medidas estatísticas delas. Apesar dos resultados promissores, acredita-se que uma caracterização mais eficiente das redes complexas pode melhorar o desempenho em relação às medidas estatísticas clássicas. Além disso, para a modelagem da rede complexa, esses métodos têm como desvantagem a necessidade de ajustar quatro parâmetros. Em resumo, as principais desvantagens presentes em métodos existentes na literatura são (considerando todas as categorias): (i) forte ênfase na aparência ou no movimento e má combinação dos dois aspectos; (ii) grandes vetores de características; (iii) algoritmos complexos e computacionalmente caros; e (iv) muitos parâmetros para ajustar.

Para superar as questões acima, nesta seção, duas abordagens desenvolvidas nesta tese para análise de texturas dinâmicas são apresentadas. Em um primeiro estudo descrito na [Subseção 5.3.2](#), uma abordagem é desenvolvida usando as RNNs para aprender representações diretamente das texturas dinâmicas (ou seja, da intensidade dos pixels), demonstrando que um método poderoso para análise de texturas estáticas (JUNIOR; BACKES, 2016) pode ser adaptado com sucesso para análise de textura dinâmica. Esse trabalho foi realizado em colaboração com o prof. Jarbas Joaci Sá Junior e foi publicado em (Sá Junior; RIBAS; BRUNO, 2019). Inspirado pelos promissores resultados alcançados nesse trabalho, uma segunda abordagem foi desenvolvida. Na [Subseção 5.3.1](#) é apresentado a segunda abordagem que visa justamente combinar uma descrição baseada em redes complexas a partir da modelagem direcionada apresentada na [Seção 4.3](#), e uma representação aprendida a partir das RNNs. Nessa abordagem, é demonstrado que além de uma modelagem mais otimizada, as RNNs também podem ser utilizadas para melhorar a caracterização das redes complexas e texturas dinâmicas. Esse estudo foi realizada em colaboração com os professores Jarbas Joaci Sá Junior e Antoine Manzanera e está publicado em Ribas *et al.* (2022). Na [Subseção 5.3.3](#), os experimentos, resultados e discussões dos métodos desenvolvidos para texturas dinâmicas são apresentadas.

5.3.1 Abordagem 1: Aprendizado de Representações de Três Planos Ortogonais (RNNDT)

Esta seção descreve a abordagem proposta para análise de textura dinâmica baseada em representações aprendidas por RNNs, nomeada como *Randomized Neural Network for Dynamic Texture* (RNNDT). A [Figura 29](#) ilustra as principais etapas do método. Primeiramente, o vídeo de textura dinâmica é dividido em três planos ortogonais, como pode ser visto na [Figura 29](#) (a). Para cada plano ortogonal, matrizes de vetores de entrada X e seus respectivos rótulos Y são construídos a partir de cada fatia. Essas matrizes de entrada e saída são submetidas as RNNs, e os pesos da camada de saída são usados como representação da fatia ([Figura 29](#) (b)). Em seguida, a média dessas representações é a representação do plano ortogonal. Por último, a representação final é a concatenação das três representações do plano ortogonal, conforme mostrado na [Figura 29](#) (c). A abordagem desenvolvida apresentada nesta seção foi publicada em

(Sá Junior; RIBAS; BRUNO, 2019).

Figura 29 – Principais etapas do método proposto RNNDT: (1) conjunto de matrizes de vetores de entrada e rótulos para cada plano ortogonal; (2) treinamento da rede neural randomizada; (3) os pesos médios da camada de saída para cada plano ortogonal são concatenados.

Fonte: Sá Junior, Ribas e Bruno (2019).

Planos ortogonais

A estratégia de três planos ortogonais é uma forma bem estabelecida e eficiente de analisar as características de aparência e movimento de vídeos de texturas dinâmicas (ZHAO; PIETIKAINEN, 2007; GONÇALVES; BRUNO, 2013; TIWARI; TYAGI, 2016b). Basicamente, o vídeo é dividido em três planos ortogonais, denominados XY , XT e YT . Considerando o vídeo como um cubo, os planos ortogonais são fatias no eixo vertical (YT), horizontal (XT) e no tempo (XY). Em outras palavras, no caso do plano XY , as fatias são os próprios quadros do vídeo. A ideia é que cada plano destaque características diferentes da textura dinâmica: o plano XY é responsável pelas características de aparência e os planos XT e YT descrevem as propriedades de movimento. Tais planos são ilustrados na Figura 29 (1).

Os planos podem ser descritos formalmente por meio da definição de pixels vizinhos para XY , XT e YT . No plano XY , os pixels vizinhos de um pixel i são definidos pela função de

vizinhança $v(i)^{XY}$. Essa função define um pixel j como vizinho de i se a distância Euclidiana entre eles for menor ou igual a R , e as coordenadas temporais t_i e t_j forem iguais, conforme

$$v(i)^{XY} = \{j \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R \text{ e } t_i = t_j\}, \quad (5.14)$$

onde x , y e t são as coordenadas Cartesianas dos pixels i e j .

Da mesma forma, as funções de vizinhança para os planos XT e YT são definidas como:

$$v(i)^{XT} = \{j \mid \sqrt{(x_i - x_j)^2 + (t_i - t_j)^2} \leq R \text{ e } y_i = y_j\}, \quad (5.15)$$

$$v(i)^{YT} = \{j \mid \sqrt{(t_i - t_j)^2 + (y_i - y_j)^2} \leq R \text{ e } x_i = x_j\}. \quad (5.16)$$

Representação Proposta

No método, a média dos pesos da camada de saída das RNNs treinadas com as fatias de cada plano ortogonal é proposta para caracterizar uma textura dinâmica. Para isso, primeiro é dividido cada fatia de um determinado plano ortogonal Ω em janelas conjuntas de $L \times L$ (L é ímpar). A partir dessas janelas, os vetores de entrada e rótulos são construídos: o valor do nível de cinza do pixel central i é considerado um rótulo y_i e os valores dos pixels da vizinhança formam o vetor de atributos de entrada $\vec{x}_i$. A [Figura 30](#) ilustra este passo para uma janela 5×5 de uma fatia do plano XY . Para cada fatia de um plano ortogonal, uma matriz de vetores de atributos $X_{(\Omega)}$ é construída concatenando os vetores de entrada $\vec{x}_i$ obtidos de janelas localizadas em cada posição possível dos pixels da fatia, ou seja, posições onde a janela está completamente na fatia (por exemplo, uma fatia 5×5 permite 9 posições diferentes para uma janela 3×3). O vetor de rótulos $Y_{(\Omega)}$ é composto pelos rótulos y_i associados aos pixels.

Para tentar analisar micro e macro texturas, são propostas janelas de três tamanhos diferentes (3×3 , 5×5 e 7×7) com pixels de vizinhos na borda determinados por um círculo com raio menor que $\sqrt{2}$, $\sqrt{8}$ e $\sqrt{13}$, respectivamente (mesmas janelas e raios propostos em ([JUNIOR; BACKES, 2016](#))). Porém, é importante ressaltar que a eficiência dessa estratégia depende da resolução do vídeo. O critério de ordem dos elementos $\{x_1, x_2, \dots, x_p\}$ é estabelecido de acordo com a [Figura 30](#).

Depois disso, é definido a função $\phi(\cdot)$ (sigmoide) e os valores da matriz de peso H . Geralmente, esses pesos são gerados de forma aleatória e podem ser diferentes a cada nova etapa do treinamento. No entanto, em métodos de extração de características, é importante que os valores da assinatura sejam sempre os mesmos para o mesmo vídeo. Portanto, o gerador LCG é utilizado assim como nas abordagens descritas anteriormente. Os números uniformes pseudo-aleatórios para a matriz H são obtidos tal como:

Figura 30 – Exemplo de extração de um vetor de atributos de entrada e seu respectivo rótulo de uma janela de uma fatia do plano ortogonal XY para treinar a RNN.

Fonte: Sá Junior, Ribas e Bruno (2019).

$$U(n+1) = (a * U(n) + b) \bmod c, \quad (5.17)$$

em que U é a sequência de números aleatórios e os valores a , b e c são parâmetros. A sequência U tem comprimento $E = Q(p+1)$, em que p e Q são o número de atributos do vetor de entrada $\vec{x}_i$ e o número de neurônios ocultos, respectivamente. Essa sequência é iniciada por $U(1) = E+1$, e os valores dos parâmetros são $a = E+2$, $b = E+3$ e $c = E^2$ (valores usados em [Junior e Backes \(2016\)](#)). Então, a matriz H é composta pelo vetor U dividido em Q segmentos de $p+1$ valores. Finalmente, os valores das matrizes H e X (cada linha) são normalizados para média zero e variância unitária.

A representação é construída com base no vetor de pesos aprendidos $\vec{f} = YZ^T (ZZ^T + \lambda I)^{-1}$, em que $\lambda = 10^{-3}$. Observe que $\vec{f}$ tem comprimento $Q+1$ devido ao valor de polarização (bias). Assim, o vetor de um plano ortogonal Ω é proposto como a média μ dos vetores $\vec{f}$ obtidos de suas fatias. Desta forma, para o plano XY , a representação é dada por

$$\vec{F}_{(XY)} = \mu \left(\begin{bmatrix} \vec{f}_1 \\ \vec{f}_2 \\ \vdots \\ \vec{f}_T \end{bmatrix} \right), \quad (5.18)$$

em que T é o número de fatias do plano XY (ou seja, o número de quadros do vídeo).

Assim, para descrever as características de aparência e movimento da textura dinâmica, a representação combinado $\vec{F}(Q)$ considerando os três planos ortogonais é dada pela concatenação:

$$\vec{\phi}(Q)_L = [\vec{F}_{(XY)}, \vec{F}_{(XT)}, \vec{F}_{(YT)}]. \quad (5.19)$$

A segunda representação pode ser obtida concatenando os vetores anteriores usando diferentes valores de tamanho de janela L , de acordo com

$$\lambda_Q = [\vec{\phi}(Q)_{L_1}, \vec{\phi}(Q)_{L_2}, \dots, \vec{\phi}(Q)_{L_n}]. \quad (5.20)$$

O vetor λ_Q depende do valor de Q , que pode fornecer características diferentes para cada valor. Assim, é possível criar uma representação final $\Upsilon_{Q_1, Q_2, \dots, Q_n}$ combinando os vetores λ_Q para diferentes valores de Q , de acordo com

$$\Upsilon_{Q_1, Q_2, \dots, Q_n} = [\lambda_{Q_1}, \lambda_{Q_2}, \dots, \lambda_{Q_n}]. \quad (5.21)$$

Complexidade Computacional

Levando em consideração uma textura dinâmica de $N \times N \times N$ pixels, seja I uma de suas fatias $N \times N$ e J_i uma janela $L \times L$. Considerando que o núcleo do método é resolver a equação $\vec{f} = YZ^T(ZZ^T)^{-1}$, pode-se dividir sua análise em alguns passos fundamentais. Primeiro, o método varre $T_1 = (N - L + 1)^2$ posições de pixel para construir as matrizes X e Y de uma fatia I usando J_i . Em seguida, calcular $Z = \phi(HX)$ e transpor Z requerem aproximadamente $T_2 = Q(p + 1)(N - L + 1)^2 + Q(N - L + 1)^2$ e $T_3 = (Q + 1)(N - L + 1)^2$ operações, respectivamente. Para multiplicar YZ^T , calcular ZZ^T e obter a sua inversa, é necessário aproximadamente $T_4 = (N - L + 1)^2(Q + 1)$, $T_5 = (Q + 1)^2(N - L + 1)^2$ e $T_6 = (Q + 1)^3$ operações, respectivamente (existem métodos mais eficientes para calcular T_6). Finalmente, multiplicar YZ^T por $(ZZ^T)^{-1}$ requer cerca de $T_7 = (Q + 1)^2$ operações. Assim, é possível considerar o tempo total para calcular $\vec{f}$ como $T = \sum_{i=1}^7 T_i = O(N^2)$, uma vez que Q , L e p geralmente são menores que N e não dependem do número de pixels em uma fatia. Além disso, considerando que $\vec{f}$ é calculado para três tamanhos de janela neste método e que cada dimensão da textura dinâmica tem N fatias, é possível estabelecer que a representação proposta tem complexidade de tempo $3NO(N^2)$, ou seja, é $O(N^3)$. Vale ressaltar que algumas operações (e.g., regularização de Tikhonov, cálculo dos pesos aleatórios H e da média dos vetores de características em cada plano ortogonal, etc.) são suprimidas para simplificar a demonstração, visto que não alteram a complexidade de tempo do método.

Estudo de Parâmetros

Para compreender o impacto dos parâmetros na classificação e escolher uma representação, vários experimentos são realizados com diferentes valores de Q e usando o classificador 1NN (distância Euclidiana) nas bases de dados UCLA-50 (SAISAN *et al.*, 2001) e Dyntex++ (GHANEM; AHUJA, 2010) com esquemas de validação cruzada de 4 e 10 pastas, respectivamente. Tais

bases de dados e classificadores são descritos em detalhes no [Apêndice A](#). Nos experimentos, um intervalo esparsos de $Q \in \{9, 19, \dots, 59\}$ é utilizado a fim de aumentar a chance de encontrar um número adequado de neurônios ocultos. Assim, para realizar a comparação com outros métodos de análise de textura dinâmica, são considerados os resultados apresentados nas [Tabela 10](#) e [Tabela 11](#) e empregado o melhor conjunto de valores Q de uma base de dados para classificar a outra base de dados. Note que independente da assinatura e valor de parâmetro considerado, os resultados alcançados pelo método em ambas as bases são altos, indicando uma robustez à mudança de parâmetro. Dessa forma, para as bases de dados UCLA-8, UCLA-9 e UCLA-50, o melhor parâmetro da base de dados Dyntex++ ($\{09, 19\}$) é utilizado; e, por outro lado, para a base Dyntex++, os valores de parâmetros da UCLA-50 ($\{49, 59\}$) são adotados.

Tabela 10 – Resultados de classificação (RNNDT) para o vetor λ_Q com diferentes valores de Q .

Q	# de Atributos	Acurácia (%)	
		UCLA	Dyntex++
{09}	90	96,50 ($\pm 1,80$)	96,71 ($\pm 0,84$)
{19}	180	96,70 ($\pm 2,00$)	96,84 ($\pm 0,87$)
{29}	270	97,55 ($\pm 1,60$)	96,64 ($\pm 0,96$)
{39}	360	97,35 ($\pm 1,89$)	96,67 ($\pm 0,85$)
{49}	450	97,85 ($\pm 1,59$)	96,49 ($\pm 0,97$)
{59}	540	97,50 ($\pm 1,96$)	96,38 ($\pm 0,97$)

Fonte: [Sá Junior, Ribas e Bruno \(2019\)](#).

Tabela 11 – Comparação do método proposto RNNDT com vetor de característica Υ combinando diferentes valores de Q .

$\{Q_1, Q_2\}$	# de Atributos	Acurácia (%)	
		UCLA	Dyntex++
{09,19}	270	97,05 ($\pm 1,87$)	97,22 ($\pm 0,84$)
{09,29}	360	97,25 ($\pm 1,85$)	96,96 ($\pm 0,88$)
{09,39}	450	97,10 ($\pm 1,81$)	96,81 ($\pm 0,85$)
{09,49}	540	97,70 ($\pm 1,54$)	96,74 ($\pm 0,87$)
{09,59}	630	97,60 ($\pm 1,65$)	96,80 ($\pm 0,86$)
{19,29}	450	97,55 ($\pm 1,66$)	96,98 ($\pm 0,87$)
{19,39}	540	97,35 ($\pm 1,89$)	96,75 ($\pm 0,88$)
{19,49}	630	97,90 ($\pm 1,63$)	96,58 ($\pm 0,89$)
{19,59}	720	97,80 ($\pm 1,74$)	96,73 ($\pm 0,92$)
{29,39}	630	97,50 ($\pm 1,85$)	96,96 ($\pm 0,86$)
{29,49}	720	97,95 ($\pm 1,60$)	96,70 ($\pm 0,92$)
{29,59}	810	97,40 ($\pm 1,93$)	96,69 ($\pm 0,90$)
{39,49}	810	97,80 ($\pm 1,56$)	96,69 ($\pm 0,86$)
{39,59}	900	97,75 ($\pm 1,71$)	96,67 ($\pm 0,88$)
{49,59}	990	98,00 ($\pm 1,63$)	96,51 ($\pm 0,94$)

Fonte: [Sá Junior, Ribas e Bruno \(2019\)](#).

5.3.2 Abordagem 2: Aprendizado de Representações de Redes Complexas Espaço-Temporal (CPNN)

Esta seção descreve a abordagem desenvolvida que combina uma descrição baseada em redes complexas (usando a modelagem da [Seção 4.3](#)) e uma representação aprendida pelo modelo RNN. O objetivo é se aproveitar da habilidade das redes complexas para representar e modelar padrões como já demonstrado em vários trabalhos e da capacidade das RNNs de aprender características de texturas, como mostrado nas seções anteriores. Para tanto, usando uma DT, a RNN pode ser treinada para prever o valor do nível de cinza de um pixel a partir da dinâmica de evolução de suas propriedades topológicas locais, fornecidas por medidas das redes complexas espaço $\times$ tempo que modelam o vídeo com diferentes valores do parâmetro raio r . Assim como nas abordagens anteriores, o vetor de peso aprendido da camada de saída da RNN é usado para compor a representação do vídeo. Portanto, esta abordagem é referida como *Complex Patterns learned using Randomized Neural Network* (CPNN) e foi publicada no periódico *Applied Soft Computing* ([RIBAS et al., 2022](#)).

As contribuições desta abordagem podem ser resumidas em: (i) uma modelagem mais simples do que em trabalhos anteriores ([RIBAS; GONÇALVES; BRUNO, 2019](#); [GONÇALVES; MACHADO; BRUNO, 2015](#); [RIBAS; BRUNO, 2020](#)) e baseada em apenas um parâmetro (raio); (ii) uma representação robusta aprendida para caracterizar a rede complexa usando RNN, ao invés de usar medidas estatísticas clássicas; e (iii) um descritor de textura dinâmica que fornece acurácia e tempo de processamento competitivos em comparação com muitos métodos da literatura.

Representação Proposta

Nesta abordagem, a RNN é treinada com a evolução das características topológicas a partir de RCs que modelam os vídeos de texturas dinâmicas. Para tanto, para compor a matriz de entrada são considerados: grau de saída $\hat{k}_i$ e força de saída $\hat{s}_i$ de cada vértice da RC. Os pesos aprendidos da camada de saída formam a representação para caracterizar a textura dinâmica. As principais etapas do método proposto estão resumidas no fluxograma da [Figura 32](#).

Para criar a matriz de vetores de entrada para a RNN é utilizado a evolução da rede complexa para diferentes valores do parâmetro de modelagem r . Portanto, para cada vértice i da RC, um vetor de entrada e seu rótulo de saída correspondente são definidos como segue: os valores de grau de saída do vértice i para diferentes raios $r = \{1, 2, \dots, R\}$ são considerados como o vetor de atributo $\vec{x}_i = [\bar{k}_i^1, \bar{k}_i^2, \bar{k}_i^3, \dots, \bar{k}_i^R]$, em que R é o valor máximo do raio. O rótulo de saída é simplesmente o nível de cinza $y_i = I(i)$. Uma matriz de vetores de atributos $X_{\vec{k}}$ para o grau de saída e um vetor de rótulos Y são obtidos considerando todos os vértices da RC. O número de colunas de $X_{\vec{k}}$ (e o tamanho do vetor Y) é então o número de amostras N que corresponde a todos os pixels, mas também pode ser qualquer subamostra do vídeo. Desse modo, é possível analisar a dinâmica de evolução da topologia dos vértices que representam os pixels com diferentes níveis

de cinza. A Figura 31 (a) ilustra a etapa de criação das matrizes de vetores de atributos e rótulos de uma rede complexa espacial. Além da matriz $X_{\bar{k}}$, também é construída a matriz para o grau de saída ponderado $X_{\bar{s}}$.

Figura 31 – Construção do vetor de entrada e rótulo de uma RC espacial modelando texturas dinâmicas com diferentes valores de r e usando o grau de saída k_i .

Fonte: Ribas *et al.* (2022).

Na próxima etapa, os pesos da matriz H da camada de entrada da RNN são gerados seguindo o mesmo procedimento empregado na Subseção 5.3.1. Assim, o vetor de características que representa as texturas dinâmicas é formado pela matriz f , que se torna aqui um vetor $\vec{f}$ (já que os rótulos de saída são escalares), que é calculado como: $\vec{f} = YZ^T(ZZ^T + \lambda I)^{-1}$, de modo que $\lambda = 10^{-3}$ (Figura 31 (b)). Para caracterizar a aparência e o movimento de texturas dinâmicas, é proposto usar a RC espacial e a temporal como entradas para treinar as RNNs. Portanto, em primeiro lugar, duas redes neurais aleatórias são treinadas, cada uma com uma matriz diferente de vetores de entrada $X_{\bar{k}}^S$ e $X_{\bar{s}}^S$ extraídos da RC espacial. A partir dessas RNNs treinadas, dois vetores são obtidos: $\vec{f}_{\bar{k}}^S$ e $\vec{f}_{\bar{s}}^S$. Para a RC temporal, o mesmo procedimento é realizado e dois vetores são produzidos $\vec{f}_{\bar{k}}^T$ e $\vec{f}_{\bar{s}}^T$ usando matrizes $X_{\bar{k}}^T$ e $X_{\bar{s}}^T$, respectivamente. Assim, para representar a aparência e o movimento da textura dinâmica, a seguinte concatenação é proposta:

$$\vec{Y}(Q)_R = [\vec{f}_{\bar{k}}^S, \vec{f}_{\bar{s}}^S, \vec{f}_{\bar{k}}^T, \vec{f}_{\bar{s}}^T], \quad (5.22)$$

em que Q é o número de neurônios da camada oculta e R é o raio máximo da modelagem, ou seja, o número de raios usados para construir a matriz X .

A Figura 32 ilustra os passos para obter o vetor $\vec{\Upsilon}(Q)_R$, que é construído usando um único valor de Q e R . Uma segunda representação é proposta baseado no vetor $\vec{\Psi}(Q)_{R_1, R_2}$ que concatena os vetores $\vec{\Upsilon}(Q)_R$ para diferentes valores de R :

$$\vec{\Psi}(Q)_{R_1, R_2} = [\vec{\Upsilon}(Q)_{R_1}, \vec{\Upsilon}(Q)_{R_2}]. \quad (5.23)$$

Finalmente, a representação final $\vec{\Theta}(R)_{(Q_1, Q_2, Q_m)}$ é proposta para combinar o vetor $\vec{\Psi}(Q)_{R_1, R_2}$ usando diferentes números de neurônios:

$$\vec{\Theta}_{Q_1, Q_2, \dots, Q_m} = [\vec{\Psi}(Q_1)_{R_1, R_2}, \vec{\Psi}(Q_2)_{R_1, R_2}, \dots, \vec{\Psi}(Q_m)_{R_1, R_2}]. \quad (5.24)$$

Figura 32 – Fluxograma da abordagem proposta CPNN para análise de textura dinâmica.

Fonte: Ribas *et al.* (2022).

Estudo de Parâmetros

Em primeiro lugar, uma análise paramétrica da abordagem proposta é realizada. Para isso, é calculada uma representação de cada amostra das bases de dados UCLA-50 e Dyntex++ utilizando valores diferentes de Q e R . A classificação é realizada pelo 1NN (distância Euclidiana) em um esquema de validação cruzada com 4 e 10 pastas para as bases de dados UCLA-50 e Dyntex++, respectivamente. A Figura 33 resume a acurácia média para as bases UCLA-50 e Dyntex++ usando a assinatura $\vec{\Psi}(Q)_{R_1, R_2}$ com $Q = 4$ e diferentes valores e combinações de R . Na matriz da figura, as linhas representam os valores de R_1 e as colunas os valores de R_2 . A diagonal principal corresponde a um único valor de R usado para construir o vetor de características. Nesse último caso, a maior acurácia média, 94,51%, é atingida usando $R = 4$. Por

outro lado, ao usar dois valores de R combinados, a maior acurácia média é alcançada usando a combinação $(R_1, R_2) = \{4, 10\}$. Isso indica que a combinação de informações locais (raio pequeno) e regionais (raio grande) das redes complexas é mais discriminativo.

Figura 33 – Acurácia média para diferentes valores e combinação de um ou dois raios nas bases UCLA-50 e Dyntex++.

Fonte: Ribas *et al.* (2022).

A Figura 34 apresenta as acurácias usando a representação $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ com um e dois valores de Q na UCLA-50 e Dyntex++. Pode-se observar que, na base UCLA-50, as maiores acurácias são alcançadas usando grandes valores de Q (todas as combinações com $Q = 29$), enquanto pequenos valores de Q fornecem melhores acurácias na base de dados Dyntex++. Em um segundo experimento, a representação final $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ é testada combinando três valores diferentes de Q , como pode ser visto na Tabela 12. Observe que, à medida que são incrementados os valores de Q nas combinações, a acurácia tende a se estabilizar e depois diminuir, enquanto o tamanho do vetor de características aumenta. Nesse sentido, a combinação $\{4, 24, 29\}$ pode ser considerada como um bom equilíbrio entre o número de atributos e a acurácia em ambas as bases de dados. No entanto, é importante ressaltar que o método produziu bons resultados para um grande número de valores de parâmetros, demonstrando robustez às mudanças de parâmetros. Por exemplo, a combinação $\{4, 14\}$ produz 160 atributos e acurácias de 99,60% e 95,31% nas bases UCLA-50 e Dyntex ++, respectivamente.

5.3.3 Resultados e Discussões

Configuração Experimental

Para a classificação dos vetores de características e comparar os resultados dos métodos desenvolvidos com outros métodos da literatura, o classificador 1NN foi utilizado (com implementação do Weka (HOLMES; DONKIN; WITTEN, 1994)). Como protocolo de avaliação,

Figura 34 – Resultados de classificação para a representação $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ com um único ($m = 1$) ou dois diferentes ($m = 2$) valores de Q nas bases de dados Dyntex ++ e UCLA-50. As células diagonais correspondem a valores únicos de Q , enquanto as demais se referem as combinações de dois valores distintos de Q .

Fonte: Ribas *et al.* (2022).

Tabela 12 – Acurácia para o vetor $\vec{\Theta}_{Q_1, Q_2, \dots, Q_m}$ usando três valores de Q .

$\{Q_1, Q_2, Q_3\}$	# de Atributos	UCLA	Dyntex++
{04, 09, 14}	240	99,10 ($\pm 1,21$)	95,06 ($\pm 1,04$)
{04, 09, 19}	280	99,10 ($\pm 1,02$)	95,08 ($\pm 1,14$)
{04, 09, 24}	320	99,50 ($\pm 0,89$)	95,19 ($\pm 1,10$)
{04, 09, 29}	360	99,50 ($\pm 0,89$)	94,71 ($\pm 1,21$)
{04, 14, 19}	320	99,30 ($\pm 0,98$)	95,38 ($\pm 0,97$)
{04, 14, 24}	360	99,40 ($\pm 0,94$)	95,32 ($\pm 1,09$)
{04, 14, 29}	400	99,60 ($\pm 0,82$)	95,05 ($\pm 1,16$)
{04, 19, 24}	400	99,10 ($\pm 1,21$)	95,10 ($\pm 1,10$)
{04, 19, 29}	440	99,50 ($\pm 0,89$)	94,86 ($\pm 1,18$)
{04, 24, 29}	480	99,92 ($\pm 0,37$)	95,03 ($\pm 1,27$)
{09, 14, 19}	360	99,10 ($\pm 1,02$)	95,18 ($\pm 0,91$)
{09, 14, 24}	400	99,50 ($\pm 0,89$)	95,18 ($\pm 1,04$)
{09, 14, 29}	440	99,50 ($\pm 0,89$)	94,90 ($\pm 1,10$)
{09, 19, 24}	440	98,80 ($\pm 1,36$)	94,81 ($\pm 1,15$)
{09, 19, 29}	480	99,10 ($\pm 1,02$)	94,69 ($\pm 1,21$)
{09, 24, 29}	520	99,30 ($\pm 0,98$)	94,87 ($\pm 1,12$)
{14, 19, 24}	480	98,80 ($\pm 1,36$)	94,94 ($\pm 1,02$)
{14, 19, 29}	520	99,10 ($\pm 1,02$)	94,72 ($\pm 1,19$)
{14, 24, 29}	560	99,40 ($\pm 0,94$)	94,72 ($\pm 1,01$)
{19, 24, 29}	600	99,00 ($\pm 1,21$)	94,24 ($\pm 1,11$)

Fonte: Ribas *et al.* (2022).

o procedimento adotado em (RAVICHANDRAN; CHAUDHRY; VIDAL, 2009; GHANEM; AHUJA, 2010; TIWARI; TYAGI, 2016a) foi empregado para a UCLA-50, UCLA-9 e UCLA-8;

e adotado em [Gonçalves, Machado e Bruno \(2015\)](#) para a Dyntex++. Assim, validação cruzada com 4 pastas e 10 pastas foram utilizadas para a UCLA-50 e Dyntex++, respectivamente. As bases de dados UCLA-9 e UCLA-8 usaram *hold-out* (50% das amostras para treinamento e o restante para teste). A acurácia média e o desvio padrão foram usados para quantificar o desempenho dos métodos. O [Apêndice A](#) descreve os detalhes das bases de dados, classificadores, esquema de validação e métricas de desempenho.

Análise de Tempo de Processamento

A [Tabela 13](#) mostra os tempos de processamento necessário, em média, para calcular uma assinatura ou representação de uma única textura dinâmica para alguns métodos. É importante ressaltar que comparações de tempo devem ser realizadas com cautela, pois o tempo de processamento pode ser influenciado por diversos aspectos, tais como, a forma de programação, linguagem de programação, hardware etc. Nos testes, a média de 20 execuções de extração de características de uma única amostra foi calculada usando um processador Intel (R) Core i7 de 3,60 GHz, 64 GB de RAM e sistema operacional de 64 bits. O método proposto CPNN levou, em média, 3,07s e 2,47s para calcular a representação de uma amostra das bases UCLA e Dyntex++, respectivamente. Por outro lado, o método RNNDT levou o menor tempo, 1,35s na UCLA e 1,83s na Dyntex++. Embora o método CPNN tenha obtido o segundo menor tempo, os resultados demonstram que ele é muito competitivo em relação às outras abordagens, levando um tempo razoável para calcular as características, e alcançando alta acurácia.

Tabela 13 – Tempo de processamento computacional em segundos dos métodos propostos e outros métodos para calcular o vetor de características.

Métodos	UCLA	Dyntex++
VLBP	3,67	2,60
DPSW	49,02	34,13
DPSWNet	8,97	48,78
CNDT	17,68	16,45
Diffusion	6,68	4,94
Métodos Propostos		
RNNDT	1,35	1,83
CPNN	3,07	2,47

Fonte: Elaborada pelo autor.

Comparações e Discussões

Os resultados alcançados na tarefa de classificação pelos métodos desenvolvidos foram comparados com métodos da literatura. Para os métodos propostos, os resultados obtidos nas seções de estudo do método e parâmetros foram considerados. A [Tabela 14](#) apresenta as acurácias obtidas para diferentes métodos na base UCLA-50. Nesses experimentos, os resultados indicam que o método proposto CPNN atingiu a maior acurácia (99,92%), seguido pela abordagem CNN-

GoogLeNet ([ANDREARCZYK; WHELAN, 2018](#)) com uma acurácia de 99,50%. A próxima acurácia de 97,15% é alcançada pelo método proposto RNNDT.

Tabela 14 – Comparação na base de dados UCLA-50 (classificador 1NN e validação cruzada com 4 pastas). Para os métodos com '*', os resultados são retirados de ([TIWARI; TYAGI, 2016a](#)), ([TIWARI; TYAGI, 2017](#)) e ([GONÇALVES; MACHADO; BRUNO, 2015](#)), os resultados com ⁺ são obtidos do artigo original, enquanto os demais foram gerados.

Métodos	Acurácia (%)
KDT-MD* (CHAN; VASCONCELOS, 2007)	89,50
DFS* (XU et al., 2011)	89,50
3D-OTF* (XU et al., 2012)	87,10
CVLBP* (TIWARI; TYAGI, 2016b)	93,00
HLBP* (TIWARI; TYAGI, 2016a)	95,00
MEWLSP* (TIWARI; TYAGI, 2017)	96,50
LBP-TOP* (ZHAO; PIETIKAINEN, 2007)	94,50
VLBP* (ZHAO; PIETIKÄINEN, 2006)	77,50 ($\pm 8,98$)
DPSW (GONÇALVES; BRUNO, 2013)	94,60 ($\pm 1,98$)
CNN-GoogLeNet ⁺ (ANDREARCZYK; WHELAN, 2018)	99,50
CNDT ⁺ (GONÇALVES; MACHADO; BRUNO, 2015)	95,00 ($\pm 5,19$)
Diffusion ⁺ (RIBAS; GONÇALVES; BRUNO, 2019)	98,50 ($\pm 3,37$)
DPSWNet (RIBAS; BRUNO, 2020)	97,97 ($\pm 0,44$)
Métodos Propostos	
RNNDT	97,10 ($\pm 0,77$)
CPNN	99,92 ($\pm 0,37$)

Fonte: Elaborada pelo autor.

A [Tabela 15](#) fornece a comparação de métodos da literatura nas bases UCLA-8 e UCLA-9, que consideram vídeos capturados de diferentes pontos de vista na mesma classe. Na UCLA-8, o método proposto CPNN obteve a segunda maior acurácia (98,94%), que é muito semelhante à maior acurácia alcançada pelo modelo CNN-GoogLeNet (99,02%). Já o método RNNDT obteve um desempenho inferior de 97,74%. Por outro lado, na base UCLA-9, o método DPSWNet ([RIBAS; BRUNO, 2020](#)) obteve a maior acurácia (99,10%) enquanto os métodos propostos CPNN e RNNDT atingiram 98,19% e 98,54%, respectivamente.

A [Tabela 16](#) apresenta os resultados da comparação para a base de dados Dyntex++, em que os métodos propostos foram superados pelo CNN-GoogLeNet ([ANDREARCZYK; WHELAN, 2018](#)), padrões binários locais (LBP-TOP) ([ZHAO; PIETIKAINEN, 2007](#)) e VLBP ([ZHAO; PIETIKÄINEN, 2006](#)). Logo, é possível afirmar que os resultados não são estatisticamente superiores aos demais nessas duas bases de dados. No entanto, é importante enfatizar alguns aspectos dos métodos. As CNNs estão atualmente entre as abordagens mais poderosas em análise de imagens e possuem um alto custo de processamento computacional. Além disso, elas exigem um grande número de amostras para treinamento, o que é uma desvantagem em comparação aos métodos desenvolvidos. As abordagens propostas produzem vetores de características menores (CPNN: 480 atributos, mas a combinação {4,14} produz 160 atributos com

acurácias competitivas; RNNDT: 270 atributos) quando comparado ao LBP-TOP (768 atributos), VLBP (16 384 atributos), MBSIF (6 144 atributos) e MEWLSP (1 536 atributos). O tamanho do vetor de características pode ser crucial em algumas aplicações que requerem baixo custo computacional para classificar a DT e baixo consumo de memória.

Tabela 15 – Comparação dos métodos propostos com outros métodos de textura dinâmica nas bases UCLA de 9 e 8 classes (classificador 1NN e metade das amostras para treinamento e o restante para teste). Os resultados com * são retirados de (TIWARI; TYAGI, 2016a) e (TIWARI; TYAGI, 2017), os resultados com + são obtidos do artigo original, enquanto os demais foram gerados.

Métodos	Acurácia (%)	
	UCLA-9	UCLA-8
3D-OTF* (XU <i>et al.</i> , 2012)	96,32	95,80
CVLBP* (TIWARI; TYAGI, 2016b)	96,90	95,65
HLBP* (TIWARI; TYAGI, 2016a)	98,35	97,50
MEWLSP* (TIWARI; TYAGI, 2017)	98,55	98,04
MBSIF* (ARASHLOO; KITTLER, 2014)	98,75	97,80
High level feature* (WANG; HU, 2015)	92,67	85,65
DNGP* (RIVERA; CHAE, 2015)	98,10	97,00
WMFS* (JI <i>et al.</i> , 2013)	96,95	97,18
Chaotic vector* (WANG; HU, 2016)	85,10	85,00
VLBP* (ZHAO; PIETIKÄINEN, 2006)	96,30	91,96
LBP-TOP* (ZHAO; PIETIKÄINEN, 2007)	96,00	93,67
CNN-GoogLeNet ⁺ (ANDREARCZYK; WHELAN, 2018)	98,35	99,02
DPSW (GONÇALVES; BRUNO, 2013)	96,33 (± 2,46)	93,41 (± 6,01)
CNDT (GONÇALVES; MACHADO; BRUNO, 2015)	95,61 (± 2,72)	94,32 (± 4,18)
DPSWNet ⁺ (RIBAS; BRUNO, 2020)	99,10 (± 0,86)	96,55 (± 7,13)
Diffusion ⁺ (RIBAS; GONÇALVES; BRUNO, 2019)	97,80 (± 1,53)	96,22 (± 4,80)
Métodos Propostos		
RNNDT	98,54 (± 1,56)	97,74 (± 2,99)
CPNN	98,19 (± 2,27)	98,94 (± 1,42)

Fonte: Elaborada pelo autor.

Os resultados dos métodos propostos também foram comparados com outras abordagens da literatura que são mais semelhantes às propostas. Os métodos CNDT, DPSWNet e Diffusion empregam diferentes formas de modelar a textura dinâmica em redes complexas, além de usar medidas estatísticas ou caminhadas aleatórias para caracterização. No entanto, ressalta-se que tais modelagens são realizadas com quatro parâmetros, enquanto a abordagem CPNN usa somente um. Em relação a esses métodos, com exceção da base UCLA-9, o método CPNN obteve acurácias maiores em todas as bases de dados. Esses resultados indicam que a combinação da descrição baseada em uma rede complexa direcionada e representação aprendida pelas RNNs pode melhorar a caracterização da textura dinâmica.

Em relação à alguns aspectos dos métodos baseados em redes neurais. Primeiramente, ambos os métodos utilizam uma RNN com aprendizagem offline, mas acredita-se que não os limitam, já que existem trabalhos que estendem esse tipo de rede neural para aprendizado online

(PRATAMA *et al.*, 2017). Isso sugere que os métodos propostos podem ser empregados em aplicações de tempo real. Em segundo lugar, as abordagens propostas possuem duas vantagens quando comparadas às CNNs: (i) as CNNs requerem um grande número de amostras para serem treinadas, por outro lado, nas abordagens desenvolvidas, isso não é uma desvantagem, visto que cada "vídeo" é a fonte do conjunto de treinamento e mesmo pequenos vídeos fornecem grandes conjuntos de dados de treinamento (por exemplo, um vídeo de $50 \times 50 \times 50$ fornece um conjunto de treinamento de 110 592 (usando uma janela 3×3) vetores de atributos de entrada e respectivos rótulos para o RNNDT e 125 000 para o CPNN); (ii) as representações propostas podem ser usados em vários classificadores, os quais podem ser escolhidos com base em vários critérios (rapidez, simplicidade, robustez etc.).

Por fim, no que diz respeito aos métodos desenvolvidos, o RNNDT é um método que compartilha com o CPNN o mesmo núcleo fundamental que é usar pesos neurais como descritores. Ambos os métodos também usam a mesma arquitetura de rede neural (RNN), mas em um esquema de três planos ortogonais para treinar o modelo no RNNDT e com informações topológicas de redes complexas no CPNN. Enquanto o método RNNDT obteve uma acurácia ligeiramente maior nas bases de dados UCLA-9 e Dyntex++, a abordagem CPNN melhora a acurácia em 2,95% e 1,2% quando comparado com o método RNNDT nas bases UCLA-8 e UCLA-50, respectivamente. Isso indica que em alguns casos a combinação de redes complexas com RNN pode fornecer um desempenho superior. Assim, à luz das altas acurácias obtidas pelos métodos, suas representações com número reduzido de atributos e sua complexidade de tempo relativamente baixa, pode-se afirmar que as representações propostas atingem bom desempenho nesses três aspectos importantes e, portanto, é comparável aos métodos mais discriminativos do estado da arte presentes na literatura.

Tabela 16 – Comparação do método proposto com outros na base de dados Dyntex++ (classificador 1NN e validação cruzada com 10 pastas). Os resultados com ⁺ são retirados do artigo original, enquanto os outros foram gerados.

Métodos	Acurácia (%)
VLBP (ZHAO; PIETIKÄINEN, 2006)	96,14 ($\pm 0,77$)
LBP-TOP (ZHAO; PIETIKAINEN, 2007)	97,72 ($\pm 0,43$)
CNN-GoogLeNet ⁺ (ANDREARCZYK; WHELAN, 2018)	98,58
DPSW (GONÇALVES; BRUNO, 2013)	91,39 ($\pm 1,29$)
CNDT (GONÇALVES; MACHADO; BRUNO, 2015)	83,86 ($\pm 1,40$)
DPSWNet ⁺ (RIBAS; BRUNO, 2020)	93,50 ($\pm 1,27$)
Diffusion ⁺ (RIBAS; GONÇALVES; BRUNO, 2019)	93,80 ($\pm 1,08$)
Métodos Propostos	
RNNDT	96,51 ($\pm 0,94$)
CPNN	95,03 ($\pm 1,27$)

Fonte: Elaborada pelo autor.

5.4 Considerações Finais

Este capítulo apresentou as pesquisas desenvolvidas que utilizam redes neurais randomizadas para aprender representações de três categorias de texturas: em nível de cinza, coloridas e dinâmicas. Nesses três casos, abordagens inovadoras que possuem como núcleo o aprendizado de representações a partir da topologia de redes complexas foram investigadas. Isto é realizado por meio da adoção de um modelo que mapeia os dados de textura, seja ele imagem ou vídeo, em redes complexas direcionadas capazes de capturar com eficiência as características intrínsecas da textura. A partir disso, informações topológicas podem ser aprendidas pela RNN, que possui um algoritmo de aprendizado simples e rápido, produzindo, assim, representações por meio do vetor de pesos treinados da camada de saída.

Os resultados alcançados em diferentes experimentos realizados em diversas bases de dados de texturas, levam a conclusão que as abordagens produzem representações altamente discriminativas. Comparações com métodos da literatura também mostram que os métodos desenvolvidos são competitivos em diferentes aspectos, tais como, acurácia de classificação, tempo de processamento, tamanho do vetor de atributos e robustez à transformações. Respalado pelo desempenho médio geral, pode-se afirmar que as abordagens desenvolvidas são ferramentas poderosas para caracterização de texturas, cumprindo assim com um dos principais objetivos desta tese. Além disso, tais abordagens também se apresentam como uma boa opção em diversas aplicações, como demonstrado no [Capítulo 8](#). Por fim, essas observações motivam investigações futuras que explorem modelos de redes complexas e técnicas de aprendizado para reconhecimento de padrões.

ANÁLISE DE FORMAS COM REDES COMPLEXAS E RNN

Este capítulo é referente aos estudos de teoria de redes complexas e a sua combinação com conceitos de transformada da distância e redes neurais artificiais para análise de contorno de formas. Forma é um atributo visual clássico e importante para a caracterização de objetos, com os primeiros estudos datados dos anos 60 ([BACKES; FLORINDO; BRUNO, 2012](#); [BACKES; BRUNO, 2010a](#)). O interesse nesse tipo de atributo é inspirado em sistemas biológicos, em que a correspondência para formas em reações bioquímicas é necessária ([COSTA; JR, 2000](#)).

Ao longo dos anos, diferentes abordagens foram propostas para representar formas, podendo ser classificadas em três categorias principais: técnicas baseadas em esqueletos, em regiões e em contornos ([LONCARIC, 1998](#)). A categoria de um método é atribuída com base em como os atributos são extraídos da forma ([WANG *et al.*, 2014](#)). Métodos baseados em esqueleto, por exemplo, usam os eixos mediais da forma para extrair seus atributos. Esses métodos têm como vantagem a robustez para formas com oclusão e articulação. [Backes e Bruno \(2010b\)](#) propuseram um método nessa categoria que usa a dinâmica de evolução de grafo, gerada do esqueleto, para obter robustez. Os autores criam uma representação da forma por uma composição da dimensão fractal multi-escala, grau mínimo, médio e máximo de cada grafo produzido por um conjunto de limiares. Outros exemplos dessa categoria são encontrados em ([CORNEA; SILVER; MIN, 2007](#); [BIASOTTI *et al.*, 2006](#); [BAI; LATECKI, 2008](#)). Técnicas baseadas em região, por outro lado, focam em uma análise global da imagem para extrair características (por exemplo, momentos de Zernike ([ZHENJIANG, 2000](#)) e métodos baseados em momentos de Hu ([HU, 1962](#); [LIAO; PAWLAK, 1996](#))). Embora consigam caracterizar bem formas genéricas, técnicas dessa categoria não conseguem distinguir objetos que são muito semelhantes, como diferentes espécies de folhas.

As metodologias baseadas em contorno, consideradas neste capítulo, usam apenas as informações de contorno da forma para extrair características. Alguns exemplos dessa categoria

incluem descritores de Fourier (WALLACE; WINTZ, 1980), *Curvature Scale-Space* (MOKHTARIAN; BOBER, 2013) e dimensão fractal multi-escala (PLOTZE *et al.*, 2005; TORRES; FALCAO; COSTA, 2004). A maioria desses métodos consideram o contorno como um conjunto ordenado de pontos conectados. No entanto, normalmente, as técnicas baseadas em contorno são prejudicadas quando a silhueta não está completa, além de que a falta de pontos ou oclusão de uma região da forma pode afetar os resultados (BACKES; MARTINEZ; BRUNO, 2010). Para evitar essa desvantagem, Backes, Casanova e Bruno (2009a) propõem o uso de redes complexas. A introdução de RCs para modelar formas permite que sejam tratados contornos não perfeitos que sofreram degradação e são frequentemente encontrados na natureza.

Inspirado no trabalho proposto por Backes, Casanova e Bruno (2009a), que modela os pixels de contorno como uma rede complexa, duas abordagens foram desenvolvidas para análise de contorno de forma. Primeiramente, a Seção 6.1 descreve a abordagem de redes de transformada da distância (em inglês, *Distance Transform Network* (DTN)) proposta para análise de formas. Nessa abordagem, a capacidade de modelagem e análise das redes complexas é combinado com o poder de análise de complexidade da transformada da distância. Em seguida, a Seção 6.2 apresenta uma nova abordagem de análise de formas baseada em aprendizado de representações da dinâmica de evolução de redes complexas. Experimentos foram realizados com os métodos desenvolvidos e demais métodos comparados em dois *benchmarks* de formas genéricas: Kimia e ETH-80. Os métodos também foram aplicados na análise de formas naturais, que são desafiadoras devido à diversidade encontrada na natureza. Objetos naturais como plantas contêm uma variedade de diferenças dentro da mesma classe, por exemplo, a mesma espécie pode conter folhas menores ou maiores. Adicionalmente, propriedades desejáveis dos métodos foram avaliadas, tais como, tolerância a ruído, invariância à escala e invariância à rotação. Os experimentos e resultados são detalhados na Seção 6.3.

6.1 Redes de Transformada da Distância para Análise de Formas (DTN)

Esta seção apresenta uma nova abordagem para a extração de características de formas que foi publicada em Ribas, Neiva e Bruno (2019) e contou com a colaboração da doutoranda Mariane Barros Neiva. Diferentemente do método proposto por Backes, Casanova e Bruno (2009a), neste trabalho, a transformada da distância Euclidiana (em inglês, *Euclidean Distance Transform* (EDT)) é aplicada na imagem de contorno e, em seguida, os pixels que pertencem a um dado raio de dilatação r são modelados como uma RC. As características de diferentes raios de dilatação são usadas para descrever a forma. Portanto, o método proposto adiciona informações sobre a propagação de ondas da EDT no contorno. A EDT calcula a distância mínima entre um pixel da imagem e uma região de interesse (ROSENFELD; PFALTZ, 1966). Esse é um método bem conhecido e amplamente utilizado em visão computacional, análise

de formas e reconhecimento de padrões (FABBRI *et al.*, 2008). A EDT tem uma forte ligação com a morfologia matemática e pode ser entendida como uma série de dilatações consecutivas. Dessa forma, a razão deste estudo é combinar redes complexas e a EDT com o objetivo de criar um método que se aproveite das vantagens de ambas as abordagens. Embora as redes tenham se mostrado robustas e poderosas para análise de formas, a EDT de um contorno pode conter informações ricas, pois contém todas as informações da propagação de ondas e colisões (FABBRI *et al.*, 2008).

Neste contexto, esta seção descreve o método proposto para análise de formas, chamado rede de transformada da distância (em inglês, *Distance Transform Network* (DTN)). Em resumo, a EDT é aplicada sobre a imagem binária do contorno, como apresentado na Figura 35 (a)-(b). No esquema, os pixels que pertencem a um determinado valor de raio de dilatação r são modelados como uma rede. Em seguida, um conjunto de limiares é aplicado para transformar a rede regular em uma rede complexa. Assim, dada uma rede t -scala, os graus médio e máximo são calculados para compor o vetor de características, essa etapa é descrita na Figura 35 (c). Finalmente, para a representação da forma, medidas de grau para diferentes valores de limiares e raios de dilatação são usadas para obter o vetor de características final (veja a Figura 35 (d)).

6.1.1 Transformada da Distância Euclidiana

A transformada da distância Euclidiana tem sido usada em diversas tarefas de visão computacional (FABBRI *et al.*, 2008). Basicamente, a EDT de uma imagem binária I , calcula a distância mínima dos pixels de fundo B para um objeto C no primeiro plano. Assim, a saída da EDT é denominada mapa de distâncias S . Esse mapa é uma matriz 2D com valores r e cada conjunto de $S_r \in S$ contém todos os pixels com uma distância mínima r para o ponto mais próximo no objeto C (o contorno), isto é, S_r tem todos os pixels com o mesmo raio de dilatação do objeto de interesse.

Os raios de dilatação formam isolinhas (pixels com o mesmo valor de r) que são uma importante fonte de análise no mapa de distâncias. Essas isolinhas contêm informações importantes sobre a propagação de ondas da EDT e, conseqüentemente, da complexidade do contorno. O "choque" entre as ondas está diretamente associado com características específicas do objeto de interesse, tais como, curvas e tamanho. Portanto, cada conjunto de pixels de um raio de dilatação, $S_r \in S$, pode ser analisado separadamente, sendo entendido como uma escala diferente. Conseqüentemente, as características de diferentes "escalas" podem ser combinadas para obter assinaturas mais robustas. A fim de reduzir a complexidade computacional, este trabalho utiliza uma implementação linear da EDT exata da referência (MAURER; QI; RAGHAVAN, 2003) e disponível no software Matlab 9.2.

Figura 35 – Sumarização do método proposto DTN de caracterização de formas baseado em transformada da distância e redes complexas.

Fonte: Ribas, Neiva e Bruno (2019).

6.1.2 Modelando Raios de Dilatação como Redes

Nesta seção, a proposta de modelar os raios de dilatação como uma rede complexa é descrita. Primeiro, a EDT é aplicada considerando o contorno C como objeto de interesse (Figura 35 (a)). A saída da EDT é um mapa de distância S mostrado na Figura 35 (b). Esse mapa S é composto por vários raios de dilatação formados por pixels que estão a uma mesma distância

r do contorno. O objetivo deste estudo é avaliar como um objeto progride ao longo de diferentes raios de dilatação (distâncias), analisando cada raio r separadamente e então combinando suas características para uma classificação robusta. Cada raio r é composto por um conjunto de pontos (ou pixels) $S_r \in S$, sendo cada ponto endereçado como $s_i = [x_i, y_i]$ ($s_i \in S_r$), em que x_i e y_i são valores discretos que correspondem às coordenadas do ponto i no mapa.

É importante ressaltar que neste estudo os pontos de um raio de dilatação, S_r , são conectados para criar a rede complexa, ao invés dos pixels do contorno como no trabalho anterior de Backes, Casanova e Bruno (2009a). Desse modo, a fim de utilizar a teoria de redes complexas no problema, dado um subconjunto $S_r \in S$, este é modelado como um grafo $G_r = (E_r, V_r)$. Assim, diferentes redes relacionadas à propagação de ondas da EDT são criadas e adicionadas à análise. Portanto, um grafo é construído, no qual cada ponto s_i de um raio de dilatação r é representado como um vértice $v \in V_r$ (ou seja, $S_r = V_r$). O conjunto de arestas não direcionadas $E_r : V_r \times V_r$ é definido pela conexão de todos os vértices em V_r entre si. Para cada aresta $e_{ij} \in E_r$ (na qual os vértices i e j estão conectados), um peso não negativo w_{ij} é definido:

$$w_{ij} = d(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (6.1)$$

Assim, a rede é representada por uma matriz de peso W_r de tamanho $N \times N$,

$$W_r([w_i, w_j]) = d(s_i, s_j). \quad (6.2)$$

Adicionalmente, para fins de invariância, esta matriz de peso é normalizada no intervalo $[0, 1]$, conforme o maior peso:

$$W_r = \frac{W_r}{\max_{w_{ij} \in W_r}}. \quad (6.3)$$

Em um primeiro estágio, todos os vértices que modelam um determinado raio r são conectados formando um comportamento regular. No entanto, uma rede regular não possui nenhuma propriedade topológica e não é considerada uma rede complexa. Portanto, é necessário transformá-la a fim de destacar propriedades importantes que caracterizam o problema estudado. Essa transformação é apresentada a seguir.

6.1.3 Assinatura

Uma abordagem para transformar a rede é a aplicação de um limiar t em suas arestas, o que produz um novo conjunto de arestas E' (COSTA *et al.*, 2007). Essa transformação consiste na seleção de arestas cujos pesos são menores que um determinado limiar t . Em Backes, Casanova e Bruno (2009a), foi demonstrado que essa estratégia permite converter uma rede inicialmente regular, que modela um contorno, em uma rede complexa que apresenta propriedades de pequeno

mundo (WATTS; STROGATZ, 1998), uma característica apresentada em muitas redes reais. Dessa maneira, a estratégia usada neste trabalho é: dada uma rede que modela um raio de dilatação da EDT, conforme descrito na seção anterior, aplica-se uma transformação $\delta_t(W_r)$ com diferentes valores de t sobre a rede para, em seguida, calcular medidas de grau em cada estágio da evolução da RC para formar o vetor de características (Figura 35 (c)). Após a aplicação de um limiar t em W_r , a matriz A_t^r é obtida (BACKES; CASANOVA; BRUNO, 2009a),

$$A_t^r = \delta_t(W_r) = \forall w \in W_r \begin{cases} a_{ij} = 0 & \text{if } w_{ij} \leq t \\ a_{ij} = 1 & \text{if } w_{ij} > t. \end{cases} \quad (6.4)$$

Como mencionado anteriormente, um conjunto de limiares $t \in T$ é utilizado para transformar redes regulares sem propriedades significativas, em redes complexas que podem conter características intrínsecas do sistema. Assim, a caracterização da forma é realizada usando várias transformações $\delta_t(W_r)$ com o conjunto T . Esse conjunto é definido por um limiar inicial t_0 e incrementado em um intervalo regular t_i até um limiar final t_f , criando diferentes redes para cada raio r . A Figura 36 (b) mostra a dinâmica de evolução da rede de um raio de dilatação para diferentes valores de limiares. Como pode ser visto, à medida que o limiar aumenta, o número de arestas nas regiões de curvas também aumenta. Uma vez aplicada a dinâmica de evolução na rede, um vetor de características é construído usando medidas de grau para a caracterização das formas.

Descritores de Grau

Para caracterizar um raio de dilatação r modelado como uma rede G_r^t , os descritores de grau são calculados a partir da matriz não ponderada A_t^r de cada rede criada com um raio r e um limiar t . O grau médio $\bar{k}$ e o grau máximo $\hat{k}$ de cada RC gerada após a aplicação do limiar são usados para compor o vetor de característica. No entanto, antes de calcular essas medidas, os graus dos vértices são normalizados em função do tamanho da rede, ou seja, do número de vértices N ,

$$\forall k_i = \frac{k_i}{N}. \quad (6.5)$$

O objetivo dessa normalização é tornar a caracterização invariante ao tamanho da RC, pois os grafos podem ter diferentes números de vértices. Para descrever a topologia da rede G_r , que modela um raio de dilatação r , é apresentado um vetor ϑ_r formado pelo grau médio $\bar{k}$ e o grau máximo $\hat{k}$. Tal vetor de características consiste na concatenação dos valores $\bar{k}$ e $\hat{k}$ de cada limiar t avaliado:

$$\vartheta_r = [\bar{k}(t_0), \hat{k}(t_0), \bar{k}(t_0 + t_i), \hat{k}(t_0 + t_i), \dots, \bar{k}(t_f), \hat{k}(t_f)]. \quad (6.6)$$

Figura 36 – Exemplos de modelagens de redes: (a) diferentes raios de dilatação r modelados como redes complexas; (b) raio de dilatação $r = 0$ modelado com uma rede transformada por diferentes valores de limiar t .

Fonte: Ribas, Neiva e Bruno (2019).

Combinando Diferentes Raios de Dilatação

A Figura 36 (a) apresenta exemplos de diferentes raios de dilatação de um contorno de forma modelados como redes complexas. Uma vez que a rede complexa é obtida a partir de um raio de dilatação, medidas de suas propriedades topológicas podem ser extraídas para reconhecimento de padrões. O vetor anterior ϑ_r calcula apenas características para um raio r . No entanto, para obter um vetor de características robusto com informações diferentes sobre o contorno da forma, vetores ϑ_r são concatenados para diferentes valores de raios de dilatação r (Figura 35 (d)). Para ilustrar tal abordagem, a Figura 37 apresenta o vetor de características ϑ_r composto pelo grau médio e grau máximo de uma forma para diferentes raios de dilatação r . Como pode-se observar, cada raio modelado contém informações distintas sobre a forma, o que aumenta o desempenho na tarefa de classificação. Para criar o vetor de características, os valores do raio variam entre o intervalo $r_0 \leq r \leq r_f$, que é avaliado na Subseção 6.1.5. Assim, um vetor combinado φ é dado por:

$$\varphi = [\vartheta_{r_0}, \vartheta_{r_1}, \vartheta_{r_2}, \dots, \vartheta_{r_f}]. \quad (6.7)$$

Esse vetor pode representar o contorno da forma considerando propriedades desejáveis

Figura 37 – Características de grau de três raios de dilatação de um mesmo contorno.

Fonte: Ribas, Neiva e Bruno (2019).

para a tarefa de classificação de forma, tais como: invariância à escala e rotação, alcançadas através da normalização do grau; e intolerância a ruído, atingida pela flexibilidade da RC.

6.1.4 Complexidade Computacional

Para avaliar o desempenho do método DTN em termos de custo computacional, uma análise da sua complexidade computacional foi realizada. Primeiramente, o método proposto aplica a transformada da distância Euclidiana binária em uma imagem binária com o contorno, isto significa que o custo computacional dessa abordagem deve ser levado em consideração. A EDT exata de uma imagem binária pode ser calculado em tempo linear de acordo com o número total de pixels de contorno (ou seja, N_p) (MAURER; QI; RAGHAVAN, 2003), que leva a uma complexidade computacional de $O(N_p)$. Assim, a próxima etapa é criar a rede complexa. Considerando um raio de dilatação com N_d pontos, cada ponto é conectado aos demais. Portanto $O(N_d + \frac{N_d(N_d-1)}{2})$ operações são necessárias para construir a rede regular. A seguir, para criar redes complexas, um conjunto de limiares T é aplicado, em que $\|T\| = n$ é o número de limiares aplicados. Os cortes nas redes complexas são executados sobre um conjunto de raios de dilatação de tamanho m . Uma vez que n e m são valores independentes de N_d e $n \ll N_d$ e $m \ll N_d$ (por exemplo, $n = 13$, $m = 49$ e $N = 5000$), pode-se ignorar n e m na complexidade, levando a uma complexidade computacional de $O(N_p + N_d + \frac{N_d(N_d-1)}{2})$.

6.1.5 Estudo de Parâmetros

Esta seção reporta uma avaliação dos parâmetros do método proposto e sua consequência no reconhecimento de formas. O método DTN assume os seguintes parâmetros: (i) conjunto de limiares T da dinâmica de evolução da rede e (ii) raio inicial e final de dilatação, r_0 e r_f . Para realizar essa tarefa, três bases de dados foram utilizadas: ETH-80, Folhas USP e Kimia. A classificação dos vetores de características nessa análise foi realizada usando o classificador

LDA e validação cruzada com 10 pastas. Os detalhes das bases de dados e classificadores são descritos no [Apêndice A](#).

Inicialmente, o comportamento do método foi analisado para diferentes conjuntos de limiares usando um raio constante de dilatação $r = 0$. A [Tabela 17](#) apresenta as acurácias para 11 configurações diferentes do conjunto de limiares ($T_1, T_2, T_3, \dots, T_{11}$). O conjunto de limiares consiste em um limiar inicial t_0 , que é incrementado através de um valor fixo t_i até um valor final t_f . Note que o método proposto alcança bom desempenho para qualquer conjunto de limiares considerado, demonstrando boa capacidade discriminativa. Como pode ser observado na tabela, não é necessário um grande número de transformações na RC para alcançar bons resultados. Os resultados também sugerem que um pequeno incremento t_i é mais apropriado para classificação. Nas três bases de dados, a melhor acurácia foi obtida pelo mesmo conjunto de limiares. De acordo com essa evidência, foi decidido adotar o conjunto de limiares T_1 ($t_0 = 0,025, t_i = 0,075, t_f = 0,950$) para todos os experimentos.

Após definir o conjunto de limiares, diferentes combinações de raios de dilatação foram testadas. A combinação de raios de dilatação usada para compor o vetor de características é definida por um raio inicial r_0 e um raio final r_f (considerando todos os raios possíveis entre r_0 e r_f). A [Figura 38](#) apresenta gráficos da acurácia em função de diferentes valores de r_0 e r_f nas bases Kimia, ETH-80 e Folhas USP, respectivamente. Note que, na maioria dos valores de raios inicial e final utilizados nas três bases de dados, o método proposto alcança bons resultados. Isso mostra que o método proposto não é muito sensível a estes parâmetros. Portanto, de acordo com os resultados, o método atinge o melhor desempenho considerando às três bases de dados para $r_0 = 0$ e $r_f = 113$. Essa análise sugere que os parâmetros encontrados para o método proposto alcançarão bons resultados em diversas bases de dados, tal como as demais avaliadas. Portanto, acredita-se que não é necessária uma busca abrangente pelos parâmetros do método proposto para diferentes bases de dados.

Tabela 17 – Resultados do método proposto para diferentes conjuntos de limiares T e usando raio de dilatação $r = 0$.

T	t_0	t_i	t_f	# de Atributos	ETH-80	Folhas USP	Kimia
T_1	0,025	0,075	0,950	26	72,43	84,00	97,00
T_2	0,025	0,100	0,925	20	71,09	83,50	97,00
T_3	0,025	0,125	0,900	16	70,03	82,16	95,00
T_4	0,025	0,050	0,500	20	72,22	79,16	92,88
T_5	0,025	0,175	0,900	12	67,71	76,16	95,00
T_6	0,050	0,150	0,950	14	68,32	81,00	91,88
T_7	0,050	0,100	0,950	20	70,51	82,83	95,00
T_8	0,050	0,075	0,800	22	70,60	79,33	97,00
T_9	0,050	0,025	0,500	38	72,16	77,66	94,00
T_{10}	0,100	0,050	0,900	34	71,00	82,16	94,88
T_{11}	0,100	0,100	0,900	18	69,35	83,00	95,00

Fonte: [Ribas, Neiva e Bruno \(2019\)](#).

Figura 38 – Resultado de reconhecimento de formas para três bases de dados como uma função do raio inicial r_0 e o raio final r_f .

(a) Kimia

(b) ETH-80

(c) Folhas USP

Fonte: Ribas, Neiva e Bruno (2019).

6.2 Aprendizado de Representações Complexas por meio de RNNs em Mapas de Características Topológicas (CRSA)

Nesta seção, um método é proposto para análise de formas baseado em representações complexas aprendidas empregando RNNs em mapas de características topológicas obtidos a partir de um esquema de modelagem de contorno em redes complexas. Neste método, o contorno da forma é modelado como uma rede complexa considerando cada ponto como um vértice e conectando todos os vértices por arestas. Em seguida, mapas de características topológicas derivados da dinâmica de evolução da RC são calculados, estes são correlacionados às características físicas do contorno da forma. As RNNs são empregadas nos mapas de características usando uma abordagem de janela deslizante para aprender as representações. Os pesos treinados da camada de saída da RNN formam uma representação eficiente que é utilizada em tarefas de classificação. Este método é denominado neste trabalho como: *Learning Complex Representation for Shape Analysis* (CRSA).

6.2.1 Modelando Contorno como Redes Complexas

Esta seção descreve a modelagem do contorno da forma como uma rede complexa, conforme proposto em Backes, Casanova e Bruno (2009a). Essa abordagem de modelagem tem sido utilizada com sucesso na análise de formas devido ao seu bom desempenho na representação e robustez às diversas transformações. O contorno da forma é um conjunto de pixels/pontos que estão em sua borda. Formalmente, considere um contorno de forma como um conjunto de N pontos de fronteira ou borda $S = \{s_1, s_2, \dots, s_N\}$, em que para cada ponto s_i são atribuídos dois valores discretos que são suas coordenadas cartesianas $s_i = (x_i, y_i)$ na imagem. Para modelar o contorno da forma como uma rede complexa $G = (V, E)$, os pontos em S são considerados vértices V , ou seja, cada ponto s_i é mapeado em um vértice v_i , criando o conjunto $V = \{v_1, v_2, \dots, v_N\}$. Para criar o conjunto de arestas E , todos os vértices são conectados entre si por uma aresta não direcionada e_{ij} , que liga os vértices v_i e v_j . Para cada aresta e_{ij} , calcula-se um peso $w_{ij} \in W$ com base na distância Euclidiana $d(e_{ij})$ entre os pontos s_i e s_j ,

$$w_{ij} = d(e_{ij}) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (6.8)$$

Para fins de invariância, os pesos das arestas são normalizados pelo peso máximo em W , tornando os pesos no intervalo $[0, 1]$,

$$w_{ij} = \frac{w_{ij}}{\max_{w_{ij} \in W}}. \quad (6.9)$$

Até o momento, observe que o grafo modelado é regular, pois todos os vértices estão conectados uns aos outros. Uma vez que um grafo regular não é uma rede complexa, uma estratégia é realizar alguma transformação nesse grafo para revelar propriedades da forma e obter um comportamento complexo. Assim, a função $\delta(E, t)$ pode ser empregada sobre as arestas para calcular um novo conjunto de arestas E_t e, conseqüentemente, revelar uma rede complexa com diferentes propriedades topológicas. Essa função consiste em remover arestas cujos pesos são iguais ou superiores a um determinado valor de limiar t , de acordo com

$$E_t = \delta(E, t) = \{e_{ij} \in E \mid w_{ij} < t\}. \quad (6.10)$$

Os valores de limiar t influenciam diretamente na estrutura e propriedade da topologia revelada na rede complexa resultante. Essas propriedades da RC estão relacionadas aos aspectos físicos e morfológicos do contorno da forma. Para pequenos valores de t , a rede complexa modela os detalhes da forma compostos por pequenos conjuntos de pontos que estão muito próximos, como uma curvatura, o que pode ser visto na [Figura 39 \(a\)](#). Por outro lado, à medida que o valor de t é incrementado, a rede complexa modela as características globais da forma, como pode ser notado na [Figura 39 \(c\)](#).

Figura 39 – Modelagem de um contorno de folha como redes complexas usando três valores diferentes de limiar t . Os pontos de contorno estão em preto e as arestas em cinza.

Fonte: Elaborada pelo autor.

Para obter uma descrição robusta da topologia da RC, é possível analisar sua dinâmica de evolução empregando diferentes transformações. Para isso, a função $\delta(E, t)$ é aplicada para diferentes valores de t e calculado as medidas dos vértices para cada RC gerada a fim de quantificar sua topologia. Assim, é usado um conjunto de limiares T composto de valores que começam a partir de um limiar inicial $0 < t_1 < 1$, sendo incrementado por uma constante t_{inc} até um limiar final $0 < t_M < 1$. Portanto, um dado valor de limiar $t_i \in T$ é determinado por $t_i = t_{i-1} + t_{inc}$ tal que $t_1 \leq t_i \leq t_M$. Esse conjunto de limiar irá derivar redes complexas que contém diferentes propriedades topológicas que mapeiam desde informações pequenas da forma até globais. Neste trabalho, duas medidas de vértice são empregadas para descrever a dinâmica de evolução da RC: grau e força (grau ponderado). Como já descrito anteriormente, o grau k_i é o

número de arestas conectadas ao vértice v_i , enquanto a força s_i é a soma dos pesos das arestas conectadas em v_i .

6.2.2 Mapas de Características Topológicas

A evolução da RC contém informações relevantes sobre os aspectos físicos do contorno da forma. Assim, a ideia é incorporar as medidas topológicas calculadas a partir da dinâmica de evolução da RC em uma representação matricial, denominada Mapa de Características Topológicas (em inglês, *Topological Feature Map* (TFM)). Nesta abordagem, dois TFMs que resumem as informações sobre a dinâmica de todos os vértices da rede são propostos: mapa de características de grau $\mathbf{F_K}$ e mapa de características de força $\mathbf{F_S}$. O mapa de características de grau é uma matriz $\mathbf{F_K}$ com os graus de vértices para vários valores de t , como segue:

$$\mathbf{F_K} = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & \cdots & t_M \end{matrix} \\ \begin{bmatrix} k_1^{t_1} & k_1^{t_2} & k_1^{t_3} & \cdots & k_1^{t_M} \\ k_2^{t_1} & k_2^{t_2} & k_2^{t_3} & \cdots & k_2^{t_M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k_N^{t_1} & k_N^{t_2} & k_N^{t_3} & \cdots & k_N^{t_M} \end{bmatrix} & \begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{matrix} \end{matrix} \quad (6.11)$$

em que as linhas representam os N vértices $\{v_1, v_2, \dots, v_N\} \in V$, as colunas correspondem aos M valores de limiares $\{t_1, t_2, \dots, t_M\} \in T$ e $k_i^{t_j}$ é o grau do vértice v_i calculado da RC transformada com o limiar t_j . Similarmente, o TFM da força é uma matriz $\mathbf{F_S}$ que armazena o valor da força dos vértices para os diferentes valores de limiares, como segue:

$$\mathbf{F_S} = \begin{matrix} & \begin{matrix} t_1 & t_2 & t_3 & \cdots & t_M \end{matrix} \\ \begin{bmatrix} s_1^{t_1} & s_1^{t_2} & s_1^{t_3} & \cdots & s_1^{t_M} \\ s_2^{t_1} & s_2^{t_2} & s_2^{t_3} & \cdots & s_2^{t_M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_N^{t_1} & s_N^{t_2} & s_N^{t_3} & \cdots & s_N^{t_M} \end{bmatrix} & \begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{matrix} \end{matrix} \quad (6.12)$$

6.2.3 Representações Propostas

Para a descrição do contorno da forma, o modelo RNN é empregado para aprender uma representação vetorial que codifica os principais padrões dos mapas de características topológicas. Para treinar a RNN, é preciso criar a matriz de vetores de atributos de entrada X e seu respectivo vetor de rótulo Y . Para isso, uma abordagem de janela deslizante é proposta para criar os vetores de atributos e rótulos a partir dos TFMs. A janela deslizante é aplicada nos valores de grau para diferentes valores de limiar, ou seja, em cada linha do TFM. Dada uma janela de tamanho η , os vetores de atributos $\vec{x}$ são construídos aplicando a janela em cada linha do mapa $\mathbf{F_K}$ e

deslizando-a entre as colunas. Por exemplo, para uma linha i em $\mathbf{F}_K$ e janela de tamanho η , os vetores de atributos são:

$$\begin{aligned}\vec{x}_i^1 &= [k_i^{t_1}, k_i^{t_2}, \dots, k_i^{t_\eta}], \vec{x}_i^2 = [k_i^{t_2}, k_i^{t_3}, \dots, k_i^{t_{\eta+1}}], \dots, \\ \vec{x}_i^{(M-\eta)} &= [k_i^{t_{M-\eta}}, k_i^{t_{(M-\eta)+1}}, \dots, k_i^{t_M}],\end{aligned}$$

e os rótulos desses vetores são $y_i^1 = k_i^{t_{\eta+1}}, y_i^2 = k_i^{t_{\eta+2}}, \dots, y_i^{M-\eta} = k_i^{t_M}$, respectivamente. Para cada linha em $\mathbf{F}_K$, esta abordagem cria $(M - \eta)$ vetores de atributos, resultando em $(M - \eta) \times N$ vetores e rótulos para compor a matriz X_K^η e o vetor de rótulos Y^η para treinar a RNN. O mesmo processo é empregado para construir a matriz de entrada X_S^η a partir dos TFMs de força $\mathbf{F}_S$. A Figura 40 ilustra o processo de criação da matriz de entrada X_K^3 e do vetor de rótulos Y^3 a partir do TFM $\mathbf{F}_K$ e usando $\eta = 3$.

Figura 40 – Ilustração da criação da matriz de entrada X_D^3 e vetor de rótulo Y^3 a partir do TFM de grau $\mathbf{F}_K$ e usando $\eta = 3$. Considerando a linha 1, os valores nas três primeiras colunas formam o primeiro vetor de entrada (caixa preta) em X_Y^3 e a coluna 4 o seu respectivo rótulo no vetor Y^3 (círculo preto). O segundo vetor de entrada é composto de valores entre as colunas 2 e 4 (caixa vermelha) com o seu respectivo rótulo como a coluna 5 (círculo vermelho). Esse processo é repetido para todas as linhas e colunas.

$$\begin{aligned}Y^3 &= [k_1^{t_4} \quad k_1^{t_5} \quad \dots \quad k_N^{t_4} \quad \dots] \\ X_K^3 &= \begin{bmatrix} k_1^{t_1} & k_1^{t_2} & & k_N^{t_1} & \\ k_1^{t_2} & k_1^{t_3} & \dots & k_N^{t_2} & \dots \\ k_1^{t_3} & k_1^{t_4} & & k_N^{t_3} & \end{bmatrix} \\ \mathbf{F}_K &= \begin{bmatrix} k_1^{t_1} & k_1^{t_2} & k_1^{t_3} & k_1^{t_4} & k_1^{t_5} & \dots & k_1^{t_M} \\ k_2^{t_1} & k_2^{t_2} & k_2^{t_3} & k_2^{t_4} & k_2^{t_5} & \dots & k_2^{t_M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \\ k_N^{t_1} & k_N^{t_2} & k_N^{t_3} & k_N^{t_4} & k_N^{t_5} & \dots & k_N^{t_M} \end{bmatrix}\end{aligned}$$

Fonte: Elaborada pelo autor.

A dinâmica de evolução das medidas dos vértices para diferentes valores de limiar apresenta curvas com comportamento característico (BACKES; CASANOVA; BRUNO, 2009a). Tal comportamento está relacionado à topologia da rede complexa e, portanto, aos aspectos físicos intrínsecos da forma. Assim, a estratégia empregada neste trabalho visa codificar os padrões de comportamento da dinâmica de evolução da RC na matriz de entrada para a RNN

aprender tais padrões. Além disso, a estratégia de janela deslizante cria grandes matrizes de treinamento, o que é interessante para treinar as redes neurais.

Uma vez que as matrizes de entrada e rótulos foram criadas para treinar a RNN, os pesos da camada oculta da rede neural precisam ser determinados. Na RNN, os pesos da camada oculta funcionam como uma projeção dos atributos de entrada e, portanto, influenciam os pesos aprendidos na camada de saída. Portanto, como nos trabalhos anteriores do [Capítulo 5](#), a matriz de peso H é gerada usando a técnica de gerador aleatório LCG ([LEHMER, 1951](#); [PARK; MILLER, 1988](#)) com sementes determinísticas. Essa estratégia garante que o método irá produzir o mesmo vetor de características (ou representação) para representar o mesmo contorno de forma de entrada, portanto, sempre é preciso usar os mesmos valores de peso aleatório na matriz H . Como em [Junior e Backes \(2016\)](#) e descrito nos métodos do [Capítulo 5](#), a sequência de números pseudo-aleatórios U que tem comprimento $L = Q * (p + 1)$ é gerada usando o LCG e os valores da matriz H e linhas da matriz X são normalizados com z -score.

Este método propõe usar o vetor de pesos treinados da camada de saída da RNN como a representação que será adotada como um vetor de características. Esse vetor de peso $\vec{f}$ é dado por $\vec{f} = YZ^T(ZZ^T + \lambda I)^{-1}$ com $\lambda = 10^{-3}$ e $\vec{f}$ tem comprimento $Q + 1$ por causa do bias. Deste modo, primeiro são treinadas duas RNNs com Q neurônios ocultos usando as duas matrizes de entrada (X_K^η e X_S^η) criadas com os mapas de características $\mathbf{F}_K$ e $\mathbf{F}_S$ e adotando o vetor de rótulos Y^η , resultando em dois vetores de características $\vec{f}_K$ e $\vec{f}_S$. Esses vetores são usadas para compor a primeira representação proposta $\vec{\phi}(Q)_\eta$ para o contorno da forma:

$$\vec{\phi}(Q)_\eta = [\vec{f}_K, \vec{f}_S], \quad (6.13)$$

em que η é o tamanho da janela usada para criar as matrizes de entrada. A [Figura 41](#) ilustra os passos do modelo proposto para calcular esta representação de forma. Como o parâmetro η influencia os vetores das matrizes de entrada e rótulos, um segundo vetor de características $\vec{\psi}(Q)_{\eta_1, \eta_2}$ é proposto para combinar as representações aprendidas com matrizes de entrada criadas usando diferentes valores de η ,

$$\vec{\psi}(Q)_{\eta_1, \eta_2} = [\vec{\phi}(Q)_{\eta_1}, \vec{\phi}(Q)_{\eta_2}]. \quad (6.14)$$

Por fim, para obter uma representação robusta, também é proposta uma representação que concatena os vetores $\vec{\psi}(Q)_{\eta_1, \eta_2}$ para diferentes números de neurônios ocultos Q para treinar as RNNs:

$$\vec{\Upsilon}_{Q_1, Q_2, \dots, Q_n} = [\vec{\psi}(Q_1)_{\eta_1, \eta_2}, \vec{\psi}(Q_2)_{\eta_1, \eta_2}, \dots, \vec{\psi}(Q_n)_{\eta_1, \eta_2}]. \quad (6.15)$$

Figura 41 – Esquema geral da proposta CRSA considerando um único valor de Q e η . (a)-(b) Um contorno de forma é modelado em RCs, que são transformadas por diferentes valores de limiar. (c) Os mapas de características topológicas são calculados a partir da evolução da RC. (d) Matrizes de entrada são obtidas dos TFMs e em (e) usadas para treinar as RNNs. Finalmente, os pesos de saída aprendidos formam a representação da forma $\vec{\phi}(Q)_\eta$.

Fonte: Elaborada pelo autor.

6.2.4 Complexidade Computacional

Para avaliar a complexidade computacional do método, pode-se considerar três etapas. Na primeira etapa, a rede complexa é criada conectando todos os N pontos do contorno entre si, sendo necessárias $T_1 = \frac{N(N-1)}{2}$ operações. Em seguida, para analisar a sua dinâmica de evolução é aplicado um conjunto de M limiares. Uma vez que M é um valor independente de N e $M \ll N$ (por exemplo, $M = 50$ e $N = 5000$), pode-se ignorar M na complexidade. Na segunda etapa, o método cria a matriz de atributos e rótulos para treinar a RNN. Para construir essa matriz, é necessário percorrer o TFM com janelas deslizantes de tamanho η , realizando $T_2 = (M - \eta)N$ operações. Por fim, as RNNs são treinadas usando as matrizes de atributos de entrada. Como já discutido na Seção 3.1, a complexidade da RNN é em função do número de vetores de atributos de entrada. Assim, neste método será de $T_3 = O((M - \eta)N)$, levando a uma complexidade computacional final de $O(2(M - \eta)N + \frac{N(N-1)}{2})$.

6.2.5 Estudo de Parâmetros

Nesta seção, os efeitos dos seguintes parâmetros do método proposto são investigados: (i) tamanho da janela η usada para construir a matriz de entrada, (ii) número de neurônios ocultos Q das RNNs, e (iii) limiar incremental t_{inc} para gerar o conjunto de limiares. Para realizar este estudo, duas bases de dados foram adotadas: ETH-80 com formas genéricas e Folhas Suecas com formas de folhas. Uma descrição detalhada dessas bases está disponível no Apêndice A. A classificação foi realizada usando o esquema de validação cruzada *leave-one-out* e o classificador LDA.

A primeira análise é sobre o impacto do tamanho da janela η na caracterização e classificação. Tal valor de parâmetro é importante porque define o tamanho dos vetores de atributos da matriz de entrada usada para treinar a RNN. Um conjunto de valores $\eta = \{3, 4, 5, 6, 7\}$ foi testado com valores fixos de $Q = 59$ e $t_{inc} = 0,015$. A Figura 42 apresenta uma matriz com acurácias para os conjuntos de dados ETH-80 e Folhas Suecas para diferentes valores de η e suas combinações. Como pode ser observado na diagonal principal, a representação $\vec{\phi}(Q)_\eta$ com apenas um valor de η obtém altas acurácias usando pequenos valores em ambos os conjuntos de dados. Isso ocorre porque pequenos valores de η produzem um número maior de vetores de atributos de entrada usados para a RNN aprender a representação da forma. No entanto, o melhor desempenho é obtido pela representação $\vec{\psi}(Q)_{\eta_1, \eta_2}$ que combina dois valores diferentes de η . Para esse caso, as melhores acurácias são alcançadas combinando um pequeno valor de η com um valor maior para ambas as bases de dados ($\eta = \{3, 7\}$ e $\eta = \{4, 7\}$ nas bases de dados ETH-80 e Folhas Suecas, respectivamente).

Figura 42 – Resultados para os vetores $\vec{\phi}(Q)_\eta$ e $\vec{\psi}(Q)_{\eta_1, \eta_2}$ usando diferentes valores de η .

Fonte: Elaborada pelo autor.

Para avaliar o número de neurônios ocultos Q das RNNs foram testados os valores $\{09, 19, 29, 39, 49, 59\}$. A Figura 43 apresenta a matriz das acurácias produzidas usando um valor Q único e dois valores combinados. Nesses testes, o método proposto gera o melhor desempenho usando dois valores combinados de Q em ambas bases. Assim, um segundo experimento foi realizado usando três valores combinados de Q , como mostra a Tabela 18. Observe que, à medida que o número de atributos gerados aumenta, a acurácia também melhora. No entanto, essa melhoria se estabiliza para vetores de atributos muito grandes. Assim, as melhores acurácias para a base de dados ETH-80 e Folhas Suecas estão usando $\{29, 49, 59\}$ e $\{9, 19, 59\}$, respectivamente.

Finalmente, experimentos foram realizados para definir o conjunto de limiares empregados para remover arestas e analisar a evolução da RC. Esse conjunto é definido por um valor inicial, de incremento e final. Como visto em trabalhos anteriores (BACKES; CASANOVA;

Figura 43 – Acurácias para o vetor $\vec{Y}_{Q_1, Q_2, \dots, Q_n}$ com combinações de valores de Q_1 e Q_2 .

Fonte: Elaborada pelo autor.

Tabela 18 – Resultados de classificação para a representação $\vec{Y}_{Q_1, Q_2, \dots, Q_n}$ calculada com diferentes valores de Q .

$\{Q_1, Q_2, Q_3\}$	# de Atributos	ETH-80 (%)	Folhas Suecas (%)
{9, 19, 29}	240	90,95	94,04
{9, 19, 39}	280	91,49	95,38
{9, 19, 49}	320	91,71	95,11
{9, 19, 59}	360	92,20	95,73
{9, 29, 39}	320	91,98	94,58
{9, 29, 49}	360	92,20	95,64
{9, 29, 59}	400	92,65	95,64
{9, 39, 49}	400	92,44	95,20
{9, 39, 59}	440	92,59	95,64
{9, 49, 59}	480	92,53	95,38
{19, 29, 39}	360	92,80	95,38
{19, 29, 49}	400	92,29	95,64
{19, 29, 59}	440	93,08	95,73
{19, 39, 49}	440	92,59	94,67
{19, 39, 59}	480	93,35	95,29
{19, 49, 59}	520	93,17	95,56
{29, 39, 49}	480	93,38	95,29
{29, 39, 59}	520	93,57	95,56
{29, 49, 59}	560	93,69	95,38
{39, 49, 59}	600	93,23	95,02

Fonte: Elaborada pelo autor.

BRUNO, 2009a), o limiar inicial é um valor pequeno próximo de 0 ($t_0 = 0,025$), enquanto o limiar final é um valor alto que está próximo de 1 ($t_M = 0,95$). Isso ocorre devido à necessidade de analisar os micro (pequenos limiares) e macro (maiores limiares) padrões dos contornos. Assim, neste experimento, diferentes valores de incremento t_{inc} são avaliados para definir o

conjunto de limiares. Este valor define o tamanho do conjunto de limiares e, consequentemente, o número de colunas dos TFMs. A Figura 44 mostra as curvas de acurácia para diferentes valores t_{inc} nas bases de dados ETH-80 e Folhas Suecas considerando os melhores parâmetros já encontrados. Observe que a acurácia diminui à medida que é aumentado o valor de incremento em ambos os conjuntos de dados. As maiores acurácias foram com o valor de $t_{inc} = 0,005$, que produziu 96,88% e 94,08% nas bases Folhas Suecas e ETH-80, respectivamente. Portanto, o valor $t_{inc} = 0,005$ foi utilizado como valor de incremento para definir o conjunto de limiar, que terá 186 valores de limiares. É importante ressaltar que valores pequenos para t_{inc} geram conjuntos maiores de limiares, permitindo realizar assim uma análise mais detalhada da evolução da RC.

Com base no estudo realizado acima, os melhores parâmetros foram $\eta = \{3, 7\}$, $Q = \{29, 49, 59\}$ e $t_{inc} = 0,005$ na ETH-80 e $\eta = \{4, 7\}$, $Q = \{9, 19, 59\}$ e $t_{inc} = 0,005$ para a Folhas Suecas. Embora ocorra diferença nos valores dos parâmetros em cada base, os resultados demonstraram que, de modo geral, o método é bastante robusto em termos de mudanças de parâmetros, produzindo altas acurácias para muitas combinações de valores. Por exemplo, se aplicado os parâmetros da ETH-80 para analisar a base Folhas Suecas a acurácia é de 96,62% (uma diferença de 0,26%), por outro lado, uma acurácia de 93,69% é obtida para a ETH-80 usando os parâmetros da Folhas Suecas, o que representa um decréscimo de apenas 0,39% em relação ao melhor resultado.

Figura 44 – Acurácias para diferentes valores de limiar de incremento t_{inc} nas bases ETH-80 e Folhas Suecas. As maiores acurácias são obtidas com $t_{inc} = 0,005$, fornecendo 96,88% e 94,08% nas bases Folhas Suecas e ETH-80, respectivamente.

Fonte: Elaborada pelo autor.

6.2.6 Interpretando as Representações Aprendidas

Como já estabelecido no campo de redes neurais, principalmente em CNNs, os pesos entre as camadas aprendem características que são relevantes para distinção das categorias do problema abordado. Por exemplo, em DCNNs é usual que as primeiras camadas aprendam filtros para extrair informações de baixo nível como cor e borda, enquanto as camadas seguintes aprendem detectar características de alto nível relacionadas ao problema estudado, tais como, formas e objetos. No entanto, compreender o funcionamento das redes neurais não é uma tarefa fácil, por isso muitos pesquisadores vêm aprofundando a investigação desta questão (ARRIETA *et al.*, 2020).

De modo a entender o funcionamento da abordagem proposta, um estudo foi conduzido para visualizar as representações aprendidas e as suas relações com várias formas. Para isso, inicialmente, formas geométricas com propriedades matemáticas conhecidas foram consideradas: círculo, elipse, triângulo, quadrado e pentágono. Primeiramente, para explicar o funcionamento do método, é preciso compreender as redes complexas geradas a partir das formas. Para tanto, a Figura 45 apresenta as formas geométricas modeladas em redes complexas, em que cada linha representa uma forma e as colunas são referentes à três valores de limiar, começando pelo menor valor (coluna 1) até o maior (coluna 3). Observe que as redes complexas do círculo possuem um padrão em que todos os vértices têm um número similar de conexões, independente do limiar aplicado. Isso acontece porque o círculo tem como principal propriedade a equidistância (raio) de todos os seus pontos em relação ao centro, não contendo assim regiões como cantos ou quinas que podem formar conjuntos super conectados de vértices. Por outro lado, os polígonos são formas geométricas delimitadas por segmentos de retas, que formam cantos com um determinado ângulo, por exemplo, no quadrado são quatro cantos com ângulo de 90° . Na modelagem, esses cantos formam conjuntos de vértices conectados entre si mesmo para limiares baixo (observe na primeira coluna da Figura 45), fazendo com que esses vértices tenham maior grau em comparação ao restante da rede. Portanto, essas características influenciam diretamente nas propriedades topológicas e dinâmica da rede complexa.

Para cada uma das formas, o método CRSA foi aplicado com $\eta = 7$ e $Q = 19$, gerando duas (grau e força) representações com 20 atributos cada. A Figura 46 mostra a visualização dos pesos aprendidos das cinco formas a partir da força dos vértices das redes complexas modeladas. Na figura, os pesos estão normalizados em $[0, 1]$ e representados em escala de cinza com as colunas indicando as formas e as linhas os pesos das conexões entre a camada oculta e a de saída. Como pode-se observar, alguns pesos específicos aprendem características diretamente relacionadas às propriedades das redes complexas. No peso da linha 18, é possível perceber que os valores estão relacionados com as informações de canto das formas. Isto é, o círculo e a elipse que não são polígonos possuem pesos com valores maiores, enquanto as demais que são constituídas por segmentos de reta possuem valores menores. Esse comportamento também é evidente no peso 9, em que o círculo e a elipse possuem menores pesos, enquanto as demais

Figura 45 – Formas geométricas modeladas como redes complexas (vértices em preto e arestas em cinza) usando três valores de limiar diferentes. Cada linha representa uma forma e as colunas são os valores de limiar de t_1 , t_2 e t_3 , em que t_1 é o menor valor, t_2 um valor intermediário e t_3 o maior.

Fonte: Elaborada pelo autor.

possuem pesos próximos de 1. Note também que a elipse possui um valor no peso 9 que está entre o círculo e os polígonos. Isso pode ser explicado pelo seu formato oval que permite a formação de dois conjuntos mais conectados de vértices, como pode ser visto na [Figura 46](#). Portanto, esses resultados sugerem que os pesos estão medindo as informações relacionadas aos cantos (ou ângulo) das formas, que são modelados na rede complexa como conjuntos de vértices superconectados.

Figura 46 – Pesos aprendidos da última camada da RNN na abordagem CRSa usando janela $\eta = 7$ e $Q = 19$. Cada linha corresponde ao peso de uma conexão entre a camada oculta e a camada de saída, ao passo que as colunas representam as formas geométricas. O valor do peso aprendido normalizado entre $[0, 1]$ é representado pela escala de cinza.

Fonte: Elaborada pelo autor.

A [Figura 47](#) apresenta a visualização dos pesos de oito amostras reais de folhas divididas em quatro classes. Nesse experimento, é mais difícil realizar uma associação direta entre as formas e os pesos aprendidos devido à complexidade das formas. No entanto, é possível fazer duas constatações: os pesos aprendidos para amostras da mesma classe são bem próximos, ao mesmo tempo, os vetores de pesos interclasses possuem distinções que diferenciam as espécies. Mesmo em espécies semelhantes visualmente, como as das colunas de 1 a 4, pode-se perceber que alguns pesos possuem valores que distinguem as classes, como pode ser observado nos

pesos 13 e 18. Assim, os resultados indicam que a abordagem proposta consegue aprender características morfológicas intrínsecas das espécies de folhas.

Figura 47 – Exemplos de pesos aprendidos usando a abordagem CRSA com janela $\eta = 7$ e $Q = 19$. Cada linha corresponde ao peso de uma conexão entre a camada oculta e a camada de saída, ao passo que as colunas representam as formas de folhas de 4 espécies (2 folhas por espécie). O valor do peso aprendido normalizado entre $[0, 1]$ é representado pela escala de cinza.

Fonte: Elaborada pelo autor.

6.3 Experimentos e Resultados

Esta seção apresenta os experimentos, resultados, análises e comparações dos métodos desenvolvidos para análise de formas e dos demais métodos.

6.3.1 Configuração Experimental

Para avaliar as abordagens propostas neste capítulo para análise de formas, os seguintes experimentos foram realizados. Primeiro, os resultados dos métodos foram comparados com outros da literatura em duas bases de dados com formas genéricas: Kimia ([SEBASTIAN; KLEIN;](#)

KIMIA, 2004) e ETH-80 (LEIBE; SCHIELE, 2003). Em seguida, os métodos são aplicados no reconhecimento de formas de folhas de plantas advindas das bases Folhas USP (BACKES; CASANOVA; BRUNO, 2009a) e Folhas Suecas (SÖDERKVIST, 2001). Os métodos também são avaliados em três diferentes condições que podem prejudicar a classificação: invariância à rotação e escala, e robustez a ruído.

A classificação dos vetores de características calculados pelos métodos é realizada usando dois classificadores bem conhecidos *Linear Discriminant Analysis* (LDA) (EVERITT; DUNN, 2001) e *Support Vector Machines* (SVM) (HEARST *et al.*, 1998) seguindo um esquema de validação cruzada *leave-one-out*. Nos experimentos de rotação e escala, as amostras processadas (ou seja, rotacionadas e escaladas) do mesmo contorno são usados para teste e contornos originais para treinamento. Isso é realizado para evitar que informações sobre as amostras de teste sejam usadas para treinamento. Os detalhes das bases de dados utilizadas, classificadores e esquema de validação estão descritos no Apêndice A.

Os resultados dos métodos propostos foram comparados com sete métodos da literatura para caracterização de contorno de formas ao longo dos experimentos. Esses métodos são:

- *Complex Network Degree (CND)*: assim como os métodos propostos, este extrator de características também modela a imagem como uma rede complexa. O método também aplica a evolução dinâmica para transformar a rede regular em redes complexas com propriedades significativas. Além disso, no CND, as características são obtidas concatenando medidas das redes como grau e grau conjunto (BACKES; CASANOVA; BRUNO, 2009a).
- *Descritores de Fourier*: este descritor calcula a transformada espectral do contorno e usa essa informação para extrair características (OSOWSKI *et al.*, 2002). Nos experimentos, os 20 coeficientes mais significativos do espectro foram usados como características da imagem.
- *Descritores de Curvatura*: o método analisa o contorno como uma curva extraíndo características como pontos máximos e mínimos que podem representar propriedades importantes da forma como mudanças nas direções, gerando 25 atributos (WU; WANG, 1993).
- *Momentos de Zernike*: simples, mas útil, essa técnica calcula um conjunto de momentos de Zernike (ordem 0 a 7) da imagem e os usam como um vetor de atributos. Os momentos de Zernike têm propriedades de invariância a rotação e representam a magnitude dos momentos complexos ortogonais da forma (ZHENJIANG, 2000).
- *Dimensão Fractal Multi-Escala*: a complexidade da forma pode ser medida pela dimensão fractal. Nesse método, o contorno é entendido como uma curva e suas alterações estão relacionadas à alta complexidade (TORRES; FALCAO; COSTA, 2004). O vetor de atributos contém os 50 pontos mais relevantes da forma para representação.

- *Análise de Segmentos*: descreve o contorno com base em estatísticas de segmentos de linhas retas. Ele considera porções de pontos contínuos e calcula o comprimento da linha reta entre os pontos extremos da porção (JUNIOR; BACKES, 2015). A média e o desvio padrão são calculados a partir de cada linha. Os diferentes segmentos são alcançados por diferentes porcentagens predefinidas do tamanho do contorno. O vetor final contém 34 atributos.
- *Descritores Angulares de RCs (ADCN)*: também baseado em redes complexas, este analisa o ângulo formado por pontos no contorno da forma (SCABINI *et al.*, 2017). Uma evolução dinâmica também é aplicada na RC de formas e os ângulos são calculados criando um histograma.

6.3.2 Classificação de Formas Genéricas

Os primeiros resultados são apresentados para as duas bases de dados com formas genéricas. Para o método CRSA, um intercâmbio de parâmetros foi realizado. Assim, na base de dados ETH-80 foi considerado o resultado com os melhores parâmetros da base Folhas Suecas ($\eta = \{4, 7\}$, $Q = \{9, 19, 59\}$ e $t_{inc} = 0,005$), enquanto na base Kimia é utilizado o resultado com os melhores parâmetros da base ETH-80 ($\eta = \{3, 7\}$, $Q = \{29, 49, 59\}$ e $t_{inc} = 0,005$). No método DTN, todos os resultados foram calculados com os parâmetros padrões definidos para o método: $T_1 = \{t_0 = 0,025, t_i = 0,075, t_f = 0,950\}$ e $r_0 = 0$ e $r_f = 113$.

A base de dados ETH-80 contém imagens de objetos naturais e feitos pelo homem, tais como, frutas, vegetais, animais e veículos. A Tabela 19 mostra as acurácias para classificação das formas das duas bases de dados genéricas e para vários métodos da literatura. Os resultados, na base ETH-80, mostram que os métodos propostos superam os demais métodos comparados, independente do classificador. Quando comparado os dois métodos propostos, os resultados são similares, com o método CRSA superando o DTN em apenas 1,34% usando o classificador LDA.

A segunda base de dados, Kimia, que também é composta de formas genéricas, produz um alto desempenho para todos os métodos avaliados neste trabalho. Apesar da simplicidade do algoritmo de análise de segmentos, ele produz uma das maiores acurácias entre todos os métodos comparados. Usando o classificador LDA, o método CRSA alcança 100,00% de acurácia, superando todos os métodos comparados. O método DTN obtém a segunda melhor acurácia junto com a técnica de análise de segmentos.

6.3.3 Reconhecimento de Formas de Folhas de Plantas

Os métodos também foram avaliados no problema de classificação de espécies de plantas através das formas das folhas. Nesta comparação, o resultado da base Folhas Suecas foi obtido usando os parâmetros da ETH-80 para o método CRSA, enquanto para a base Folhas USP os parâmetros da Folhas Suecas foram empregados. No método DTN foram considerados os seus

Tabela 19 – Resultados comparativos entre duas bases de dados formas genéricas usando os classificadores LDA e SVM.

Métodos	ETH-80		Kimia	
	LDA	SVM	LDA	SVM
Fourier	71,80	81,49	87,88	84,85
Curvatura	61,40	72,16	76,77	84,85
Zernike	82,86	88,47	94,95	96,97
M. S. Fractal	74,82	81,83	90,91	84,85
Análise de Segmentos	81,22	80,73	98,99	96,97
ADCN	85,40	91,04	97,98	90,91
CND	72,47	84,27	95,96	96,97
Métodos Propostos				
DTN	92,35	92,59	98,99	98,99
CRSA	93,69	92,32	100,00	97,98

Fonte: Elaborada pelo autor.

parâmetros padrão. A base Folhas Suecas, contendo imagens de folhas de plantas da Suécia, tem um total de 1125 contornos de folhas categorizados em 15 espécies. Conforme observado na [Tabela 20](#), os métodos propostos atingem as maiores acurácias, com destaque para o método CRSA que alcançou 96,62% com o classificador LDA. Em seguida, o terceiro melhor método é o CND com 82,93% e classificador SVM.

Por fim, o segundo experimento foi conduzido no conjunto Folhas USP, que contém características interessantes para *benchmark* de análise de formas, tais como, semelhanças entre as folhas de diferentes espécies e a alta variabilidade nas folhas de uma mesma espécie. Isso torna a tarefa de reconhecimento desafiadora, fazendo com que a base teste o poder de discriminação dos métodos de análise de forma. No total, são 600 imagens de folhas reais da flora brasileira (30 espécies de plantas). Os métodos desenvolvidos CRSA e DTN melhoram consideravelmente a acurácia em comparação aos demais usando ambos os classificadores. Por exemplo, o método DTN melhora o desempenho de reconhecimento de 88,50% para 94,33% sobre o ACDN e de 83,50% para 94,33% sobre o método de análise de segmentos usando o classificador LDA. Em relação aos dois métodos propostos, as acurácias são bem próximas nas duas bases de dados, com o CRSA obtendo as melhores acurácias com LDA e o DTN usando o SVM.

6.3.4 Análise de Invariância à Rotação e Escala

A fim de confirmar as propriedades de invariância à rotação e escala das abordagens, experimentos foram realizados em versões modificadas a partir da base Folhas USP. Na versão de rotação, as imagens foram rotacionadas em 7°, 35°, 132°, 201° e 298°, resultando em um conjunto com 3600 amostras. A versão para análise de escala contém imagens reescaladas por um fator de 125%, 150%, 175% e 200% a partir das imagens originais. A [Figura 48](#) (a)-(b) mostra alguns exemplos de formas rotacionadas e escaladas.

Tabela 20 – Acurácias das duas bases de dados de formas de folhas para os classificadores LDA e SVM.

Métodos	Folhas USP		Folhas Suecas	
	LDA	SVM	LDA	SVM
Fourier	73,33	79,00	53,42	72,35
Curvatura	76,67	83,50	56,27	73,07
Zernike	68,00	76,83	51,82	55,02
M. S. Fractal	70,67	75,67	74,75	72,35
Análise de Segmentos	82,83	85,33	80,00	81,78
ADCN	88,50	88,67	75,55	77,77
CND	83,50	86,17	73,78	82,93
Métodos Propostos				
DTN	94,33	92,33	74,13	96,35
CRSA	94,67	92,00	96,62	94,67

Fonte: Elaborada pelo autor.

Na [Tabela 21](#), os resultados de classificação usando imagens rotacionadas e escaladas confirmam a robustez e a invariância dos métodos propostos sobre esses tipos de transformações. O CRSA alcançou as maiores acurácias nos dois testes realizados e para ambos os classificadores, sendo bem próximas das obtidas com a base original. Observe também que os métodos propostos produziram acurácias semelhantes para os dois classificadores, enquanto outros métodos foram sensíveis aos algoritmos de aprendizado de máquina. Isso sugere vetores de características consistentes e, conseqüentemente, uma boa invariância à escala e rotação.

No caso dos métodos propostos, a propriedade de invariância à escala é devido às normalizações realizadas na modelagem da rede complexa, como proposto por [Backes, Casanova e Bruno \(2009a\)](#) e conservado pelos métodos desenvolvidos. A normalização da matriz de peso (distâncias) W é responsável por redimensionar as arestas, criando tolerância de escala na classificação. Assim, mesmo que as distâncias sejam maiores em amostras de maior escala, os pesos normalizados pela distância máxima superam tal limitação. Além disso, duas imagens em escalas diferentes produzem números diferentes de pontos em seu contorno. Esse problema é resolvido com a normalização do grau k_i pelo número de vértices N da rede complexa. Por outro lado, a invariância à rotação é intrínseca da rede complexa modelada, que não considera informação de direção ou ordem na sua construção, sendo a rede a mesma independente da direção da forma.

6.3.5 Análise de Robustez a Ruído

É fato que no processo de aquisição das imagens podem ser acrescentados erros no contorno do objeto. Isso ocorre porque essa tarefa não é perfeita, causando uma variedade de interferências e ruídos. Nos experimentos realizados anteriormente com a base de dados original Folhas USP, as formas são suaves. Para avaliar a robustez dos métodos em condições ruidosas, estes foram aplicados em uma versão da base Folhas USP modificada com ruído. Para tanto,

Figura 48 – Exemplo de formas de folhas da base Folhas USP em diferentes condições. (a) contornos rotacionados, (b) contornos escalados e (c) contornos com ruído.

Fonte: Backes, Casanova e Bruno (2009a).

Tabela 21 – Resultados para a versão rotacionada e escalada da base de dados Folhas USP.

Métodos	Rotacionada		Escalada	
	LDA	SVM	LDA	SVM
Fourier	67,53	77,17	63,25	85,29
Curvatura	76,31	83,33	76,92	84,46
Zernike	65,53	76,64	53,38	71,33
M. S. Fractal	58,17	68,58	65,25	68,50
Análise de Segmentos	81,47	76,22	82,04	79,75
ADCN	86,36	77,11	86,13	80,75
CND	80,61	84,28	81,46	85,04
Métodos Propostos				
DTN	89,06	88,67	92,38	93,08
CRSA	93,78	93,39	93,71	93,25

Fonte: Elaborada pelo autor.

quatro níveis de ruído foram aplicados em cada imagem, gerando quatro novos subconjuntos. A condição ruidosa vai aumentando do nível 1 ao 4. A Figura 48 (c) mostra algumas formas com diferentes níveis de ruído.

Os resultados do experimento de tolerância a ruído usando o classificador LDA são apresentados na Tabela 22. Os métodos propostos demonstraram grande capacidade de classificação de formas mesmo em contornos com alto nível de ruído, com destaque para o CRSA que atingiu as melhores acurácias em todas as condições analisadas. No caso do CRSA, mesmo com o maior nível de ruído, a acurácia ainda é competitiva, enquanto quando considerados todos os níveis

juntos, a acurácia é próxima dos dados originais. Ressalta-se também que o método Curvatura aplica um filtro passa-baixo quando é calculado, reduzindo o ruído do contorno. No entanto, ambos os métodos desenvolvidos não necessitam deste tipo de pré-processamento e ainda é capaz de obter um bom desempenho em formas ruidosas.

Tabela 22 – Comparação dos métodos propostos com outros da literatura para a base de dados Folhas USP corrompida por diferentes níveis de ruído usando o classificador LDA.

Métodos	Nível 1	Nível 2	Nível 3	Nível 4	Todos Níveis
CND	79,67	77,50	78,17	75,33	77,58
Fourier	65,50	62,50	59,33	55,67	58,62
Curvatura	72,00	69,33	64,33	61,67	68,71
Zernike	67,33	65,17	66,33	62,00	61,17
M.S Fractal	63,33	63,83	62,17	59,17	63,50
ADCN	83,33	81,50	80,17	79,67	81,12
Análise de Segmentos	75,17	71,83	66,17	69,67	69,58
Métodos Propostos					
DTN	87,33	79,50	75,17	72,50	86,54
CRSA	89,66	84,33	83,67	81,50	91,37

Fonte: Elaborada pelo autor.

6.4 Discussões e Considerações Finais

Este capítulo apresentou dois novos métodos para análise e classificação de formas, nomeados de DTN e CRSA, baseados em conceitos de teoria das redes complexas, RNNs e transformada da distância. Dado um contorno de forma, ou mesmo um subconjunto (pontos em sequência ou não) de contorno, o método DTN aplica a transformada de distância Euclidiana e modela cada raio de dilatação como uma rede complexa, caracterizando-a com medidas de grau médio e máximo da evolução da rede. Por outro lado, o método CRSA modela o contorno como rede complexa e utiliza redes neurais para aprender as características da sua topologia dinâmica.

Primeiramente, um estudo foi conduzido para avaliar o efeito de cada parâmetro dos métodos propostos no desempenho de classificação. A avaliação dos parâmetros mostrou que os métodos não são sensíveis à configuração dos parâmetros, uma vez que foram utilizados os mesmos parâmetros para todas as bases de dados no DTN ou feito um intercâmbio de parâmetros no caso do CRSA. Essa é uma característica muito importante, pois sugere que os métodos podem ser utilizados com parâmetros pré-configurados em outras bases de dados e o usuário não precisaria mais realizar esta avaliação de configuração. Isso é avaliado na aplicação do método em um problema real de classificação de peixe no [Capítulo 8](#).

Os resultados apresentados na seção anterior indicam que a forma é efetivamente representada pelos vetores de características calculados pelos métodos, superando os outros métodos comparados em termos de desempenho de classificação e também nos demais aspectos avaliados. O método DTN melhora significativamente a acurácia em comparação com os outros

métodos, por exemplo, de 85,40% (ADCN) para 92,35% na base ETH-80, de 82,83% (Análise de Segmentos) para 94,33% e de 88,50% (ADCN) para 94,33% na base de dados Folhas USP (considerando o classificador LDA). Além disso, observe que alguns métodos não possuem um desempenho uniforme nas várias bases de dados, ao passo que os métodos propostos conseguem bons resultados, independente do problema. Por exemplo, na base Folhas USP, o ADCN + LDA atinge 88,50%, enquanto na Folhas Suecas o resultado cai consideravelmente para 75,55%. Isso sugere que o método proposto funciona bem, em geral, nos diferentes conjuntos de dados tanto genéricos como naturais, enquanto os outros métodos são sensíveis às características específicas de cada problema. No caso das formas naturais, a extração de características consistentes é uma tarefa desafiadora devido às mudanças de iluminação, pose, escala etc.

Em termos de classificador, de modo geral, os métodos CRSA e DTN conseguem desempenhos semelhantes em ambos os classificadores, com exceção do DTN na base Folhas Suecas usando o LDA. Em contraste, outros métodos são mais sensíveis ao classificador, por exemplo, os resultados do ADCD diminuem de 97,98% (LDA) para 90,91% (SVM) na base de dados Kimia, e os resultados do CND diminuem de 84,27% (SVM) para 72,47% (LDA) na base ETH-80. Tais diferenças também são observadas nos experimentos com formas rotacionadas e escaladas. De fato, a escolha da combinação de descritor e classificador influencia diretamente nos resultados, pois o desempenho do classificador é afetado por diversos fatores, tais como, os tipos e qualidades de atributos extraídos, dimensão do vetor de características, parâmetros, número de amostras etc (AMANCIO *et al.*, 2014). Portanto, os resultados indicam que as representações calculadas pelos métodos propostos são robustas para os classificadores considerados, sendo necessário pesquisas adicionais para investigar o desempenho dos métodos em uma maior variedade de classificadores.

No que diz respeito aos métodos que compartilham do mesmo núcleo teórico (ACDN, CND, DTN e CRSA), que é a aplicação da teoria de redes complexas, algumas considerações podem ser feitas. Nesses métodos, a técnica de modelagem de redes complexas é similar em todos eles, entretanto, a caracterização ou a forma de trabalhar com a rede complexa são distintas. Assim, são justamente esses aspectos distintos que influenciam diretamente no desempenho de cada abordagem. No método CND, que propôs o modelo de redes complexas para formas, os autores empregaram duas medidas clássicas de redes complexas para caracterizar a dinâmica da sua evolução: grau médio e grau máximo. Em seguida, o método ACDN melhorou o desempenho de classificação propondo uma nova caracterização baseada em descritores angulares das arestas. Já o método DTN, embora adote as medidas do CND, o poder de análise das formas é potencializado pela utilização da combinação de modelagem de redes complexas com o poder de análise de complexidade da transformada da distância. Nesse caso, os raios de dilatação resultantes da DTN são modelados e analisados, ao invés da própria forma. Por fim, no método CRSA, uma nova estratégia é aplicada para caracterizar a dinâmica de evolução das redes complexas. Tal metodologia emprega RNNs para aprender representações a partir de mapas de características gerados da rede complexa modelada.

Dentre os quatro métodos baseados em redes complexas comparados, o CRSA alcança o melhor desempenho geral nas diversas análises e comparações realizadas. Isso indica que a rede complexa consegue modelar e incorporar na sua topologia a complexidade e aspectos físicos das formas, entretanto, muitas vezes essas informações não são adequadamente extraídas ou caracterizadas pela medida empregada. Por exemplo, o método CRSA aumenta significativamente a acurácia comparado ao CND de 83,50% para 94,67%, de 73,78% para 96,62% e de 72,47% para 93,69%, nas bases de dados Folhas USP, Folhas Suecas e ETH-80, respectivamente. Assim, pode-se afirmar que a proposta de aprendizado de representações fornece descritores mais discriminativos, evidenciando a importância do desenvolvimento de técnicas mais sofisticadas para caracterizar as redes complexas, proposta principal desta tese. No caso do DTN, também é notório que a adição da transformada da distância para a modelagem realça as características intrínsecas da forma, incrementando o desempenho consideravelmente em muitos casos. Ressalta-se também que ambas propostas conseguem garantir propriedades importantes na caracterização das formas: invariância à rotação e escala, e robustez a ruído. Isso ocorre devido à flexibilidade e capacidade das redes complexas em modelar as informações das formas, sem considerar a ordem dos pontos e direção. Por fim, pode-se concluir que às duas abordagens desenvolvidas demonstraram potencial e são boas opções para a aplicação em problemas de reconhecimento de padrões e análise de imagens que necessitem de caracterização de formas. O [Capítulo 8](#) aplica as abordagens desenvolvidas neste capítulo no problema de classificação de espécies de peixes por meio do contorno do otólito.

PADRÕES BINÁRIOS DE LLNA PARA CARACTERIZAÇÃO E CLASSIFICAÇÃO DE REDES COMPLEXAS

Neste capítulo, uma abordagem para classificação de redes complexas sintéticas e reais é apresentada. Diferentemente dos estudos apresentados nos capítulos anteriores que tinham como objetivo classificar redes complexas específicas de imagens, neste capítulo, o reconhecimento de padrões em redes complexas visa classificar redes de larga escala gerais, distinguindo-as de acordo com suas características intrínsecas (BANERJEE; JOST, 2008; MIRANDA; MACHICAO; BRUNO, 2016). Nos últimos anos, os conceitos de redes complexas têm sido aplicados para reconhecimento de padrões com resultados promissores em visão computacional e em outros problemas, tais como: atribuição de autoria (MACHICAO *et al.*, 2018a; MEHRI; DARONEH; SHARIATI, 2012; AKIMUSHKIN; AMANCIO; OLIVEIRA JR., 2017), reconstrução filogenética (BANERJEE; JOST, 2008; MACHICAO *et al.*, 2018b), estudo de rede de comércio internacional (XIN; ZHANG; SHAO, 2018), entre outros.

A abordagem apresentada neste capítulo é inspirada no trabalho realizado por Miranda, Machicao e Bruno (2016), que propõe uma família de autômatos de rede denominada como: autômato de rede Life-Like (LLNA). Esse método foi introduzido como uma ferramenta para análise de redes complexas voltada para aplicações de reconhecimento de padrões. Nele, os padrões de evolução temporal, do inglês *Temporal Evolution Patterns* (TEPs), formados a partir do comportamento dinâmico deste sistema não-linear, são usados para extrair características. Como o LLNA é um autômato celular cujos estados são binários, os TEPs extraídos são representados como cadeias de zeros e uns, representando os estados de suas células (vivo ou morto), por exemplo, 010100100010100111001. Miranda, Machicao e Bruno (2016) usaram um conjunto de três medidas: a entropia de Shannon, o comprimento da palavra e a complexidade de Lempel-Ziv, a fim de obter os vetores de características correspondentes, que são capazes de

caracterizar a topologia da rede.

Embora seja adequada para tarefas de reconhecimento de padrões, a assinatura de evolução dinâmica fornecida pelos histogramas das três medidas mencionadas acima, possui a limitação dos parâmetros para a construção dos histogramas. Uma vez que essas métricas são compostas de valores contínuos, é necessário definir um número ótimo de caixas do histograma, o que pode ser uma tarefa difícil e custosa, com influência no desempenho de classificação, além de ser necessário para todas as novas bases de dados. No trabalho em questão, esse parâmetro foi estabelecido empiricamente sem uma discussão aprofundada. Portanto, um conjunto de métricas mais discriminativas e sem esse problema poderia melhorar o desempenho da classificação.

Neste contexto, este capítulo apresenta uma extensão do trabalho anterior de [Miranda, Machicao e Bruno \(2016\)](#), denominada Padrões Binários de LLNA (em inglês, *Life-Like Network Automata descriptor based on Binary Pattern* (LLNA-BP)). A [Figura 49](#) ilustra os passos do método desenvolvido, que propõe uma forma melhorada de extrair vetores de características dos TEPs por meio de dicionários de padrões binários. Esse dicionário β_D é um conjunto discreto $\beta \in [0, 2^D - 1]$ de possíveis padrões binários com comprimento igual a D . Ao contrário da abordagem anterior, na abordagem proposta, o histograma de frequência do conjunto discreto de padrões binários é calculado ([Figura 49](#) (d)). Essa abordagem considera diferentes combinações de comprimento de padrões binários. Além disso, é demonstrado como essas medidas melhoraram os resultados obtidos por [Miranda, Machicao e Bruno \(2016\)](#). Basicamente, o método pode ser dividido em duas partes: extração de padrões binários do TEP e construção da assinatura do padrão binário. A [Seção 7.1](#) descreve essas duas etapas do método proposto, enquanto nas [Seções 7.2 e 7.3](#) são apresentados os experimentos e resultados para o método proposto e demais métodos em bases de dados sintéticas e reais. Este trabalho foi realizado em colaboração com a pós-doutoranda Jeaneth Machicao e resultou em um artigo publicado no periódico *Information Sciences* ([RIBAS; MACHICAO; BRUNO, 2020](#)).

7.1 Padrões Binários de TEPs

Apesar das medidas de complexidade usadas no método LLNA original alcançarem resultados promissores, os TEPs fornecem padrões ricos que podem ser explorados com técnicas mais eficientes. Esses TEPs são formados pela dinâmica do CA na rede complexa no tempo, ou seja, enquanto as colunas representam os vértices, as linhas correspondem ao tempo de evolução do autômato. Na abordagem proposta, diferentes dicionários de padrões binários são utilizados para obter diferentes informações sobre os TEPs e, assim, melhorar a caracterização da rede. O padrão binário é um descritor estatístico calculado considerando os estados de um vértice em tempos diferentes.

Para realizar a caracterização dos TEPs, dicionários de padrões binários de diferentes comprimentos D são empregados para compor um vetor de característica (assinatura) extraído

Figura 49 – Esquema de reconhecimento de padrões proposto baseado em padrões binários extraídos dos padrões de evolução temporal do LLNA (LLNA-BP). (a) Considere um problema em que as amostras são modeladas como rede complexa. (b) Para cada rede, em (c) o LLNA evolui com uma regra previamente selecionada com os estados representados por branco (vivo ou um) e preto (morto ou zero). De cima para baixo, o padrão de evolução temporal produzido pela dinâmica LLNA da rede dada. (d) Sequências de zeros e uns são concatenadas para produzir padrões binários de tamanhos diferentes, e.g., $D = 3, D = 4, \dots, D_n$, a partir do qual histogramas são obtidos, respectivamente, (e) formando, assim, o vector de características utilizando várias estratégias, (e.g., concatenando as distribuições de frequência global $\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^g$), que são então usados para (f) a tarefa de classificação de redes.

Fonte: Ribas, Machicao e Bruno (2020).

dos padrões de evolução temporal do LLNA. Formalmente, um dicionário β_D é um conjunto discreto $\beta \in [0, 2^D - 1]$ contendo todas as combinações possíveis de zeros e uns de comprimento D , isto é, β_D é o alfabeto que contém padrões binários como $\beta_D = \{00, 01, 10, 11\}$ para $D = 2$, $\beta_D = \{000, 001, 010, 011, 100, 101, 110, 111\}$ para $D = 3$ e assim por diante. Dessa maneira, um padrão binário é um descritor estatístico calculado considerando os estados de uma célula (vértice) ao longo do tempo.

Primeiro, dada uma célula c_i (vértice da rede) no tempo t , um padrão binário contendo D bits é construído. Esse padrão binário é composto pelos estados de uma célula c_i , iniciando em $s(c_i, t)$ até $s(c_i, t + (D - 1))$. Para uma melhor representação, esses padrões binários são codificados em um formato decimal $\Phi_D(c_i, t)$, como segue:

$$\Phi_D(c_i, t) = \sum_{j=t}^{(D-1)+t} s(c_i, j) 2^{(j-t)}. \quad (7.1)$$

Em seguida, um conjunto de padrões binários é produzido para cada célula c_i através de uma janela deslizante, resultando no seguinte conjunto, $\{\Phi_D(c_i, t), \Phi_D(c_i, t + 1), \dots, \Phi_D(c_i, t + (D - 1))\}$. Desta forma, considerando um tempo máximo T e tamanho de dicionário D , para cada vértice (célula) da rede, $(T - D + 1)$ padrões binários são computados, como ilustrado na

Figura 49 (d).

Neste trabalho, duas maneiras de compor os vetores de características usando o dicionário de padrões binários foram exploradas. O primeiro, chamado histograma global H_g^D , e o segundo, denominado histograma de graus H_k^D , são detalhados a seguir.

7.1.1 Histograma Global

Como o número de padrões binários possíveis é igual a 2^D , a partir de todos os padrões binários gerados por uma rede, um histograma global H_g^D é computado para caracterizá-la, como pode ser visto na Figura 49 (e). Esse histograma global calcula a ocorrência dos padrões binários no TEP da rede, de acordo com

$$H_g^D(p) = \sum_{i=1}^N \sum_{t=0}^{(T-D+1)} \delta(\Phi_D(i, t), p), \quad (7.2)$$

onde $p \in [0, 2^D - 1]$, N é o número de vértices na rede complexa e $\delta(x, y) = \begin{cases} 1, & x = y, \\ 0, & \text{caso contrário.} \end{cases}$

Na Equação 7.2, note que p é a representação decimal de um padrão binário gerado pela função $\Phi_D(i, t)$. Em seguida, o histograma $H_g^D(p)$ é então calculado como uma função de densidade de probabilidade, isto é, este é dividido pelo número total de padrões a fim de tornar o histograma invariante ao tamanho da rede, como segue:

$$H_g^D(p) = \frac{H_g^D(p)}{\sum_{i=0}^{(2^D-1)} H_g^D(i)}. \quad (7.3)$$

7.1.2 Histograma de Graus

Com relação aos valores de k , também é possível calcular histogramas de padrões binários somente para os vértices com o mesmo grau k . Esse histograma tem como objetivo obter informações da ocorrência de padrões binários em vértices similares na rede (considerando o grau). Deste modo, um histograma H_k^D é gerado para cada grau k da rede. Tais histogramas podem ser definidos da seguinte maneira:

$$H_k^D(p) = \sum_{i=1}^N \sum_{t=0}^{(T-D+1)} \delta_k(\Phi_D(i, t), p, k_i), \quad (7.4)$$

onde $p \in [0, 2^D - 1]$ e $\delta_k(x, y, k_i) = \begin{cases} 1, & x = y \text{ e } k_i = k, \\ 0, & \text{caso contrário.} \end{cases}$

Similarmente ao histograma global, o histograma de graus $H_k^D(p)$ é computado como uma função de densidade de probabilidade, dividindo pelo número total de padrões de cada grau k , de acordo com

$$H_k^D(p) = \frac{H_k^D(p)}{\sum_{i=0}^{(2^D-1)} H_k^D(i)}. \quad (7.5)$$

A Figura 50 (a) mostra os TEPs obtidos para redes aleatórias, geográficas, de pequeno mundo e livre de escala, com $N = 500$ vértices. Esses TEPs foram gerados usando a regra B135678-S03456 e foram evoluídos por $T = 500$ passos de tempo. Três exemplos para o histograma $H_g^D(p)$ usando diferentes valores de D são dados na Figura 50 (b). Para cada modelo de rede, os histogramas obtidos para duas amostras de TEPs foram plotados (Figura 50 (a)). Esses exemplos são desafiadores, pois são semelhantes visualmente. No entanto, pode-se observar na figura que os histogramas de cada modelo de rede são semelhantes, enquanto os histogramas das diferentes classes são distintos, minimizando assim a variância intraclasse e maximizando a variância interclasse, o que corrobora o poder de discriminação da abordagem proposta.

7.1.3 Assinatura

Os vetores de características (isto é, a assinatura) são construídos com base nos histogramas H_g^D e H_k^D . O primeiro vetor de características considerado é composto por histogramas globais H_g^D com valores diferentes de D . Essa estratégia visa combinar diferentes tamanhos de padrões binários para a representação da rede complexa, de acordo com

$$\vec{Y}_{D_1, D_2, \dots, D_n}^g = [H_g^{D_1}, H_g^{D_2}, \dots, H_g^{D_n}]. \quad (7.6)$$

Em seguida, uma vez que existe um histograma H_k^D para cada grau da rede, propõe-se a média μ e o desvio padrão σ de todos esses histogramas para caracterização da rede. Assim, é possível analisar a frequência dos padrões binários em vértices com diferentes graus. Por exemplo, a frequência de um determinado padrão binário pode ser alta para os vértices com baixo grau e, o oposto, para os vértices com alto grau. Dessa forma, o vetor de características de média $F(D)_\mu$ e o vetor de características de desvio padrão $F(D)_\sigma$ são obtidos, de acordo com

$$F(D)_u = \mu \left(\begin{bmatrix} H_{k_1}^D \\ H_{k_2}^D \\ \vdots \\ H_{k_K}^D \end{bmatrix} \right), \quad (7.7)$$

Figura 50 – Duas amostras de padrões de evolução temporal para quatro modelos de redes diferentes: aleatório, pequeno mundo, livre de escala e geográfico, e seus histogramas. (a) Os TEPs correspondem a redes com $N = 500$ vértices, grau médio $\langle k \rangle = 10$, sendo evoluídos com a regra B135678-S03456 durante $T = 350$ passos de tempo. Cada coluna é um modelo de rede diferente e cada linha é uma amostra de rede diferente. (b) Histogramas correspondentes ao histograma global $H_g^D(p)$ considerando diferentes valores de comprimento do padrão binário D . Cada curva corresponde a uma das amostras em (a).

Fonte: Ribas, Machicao e Bruno (2020).

$$F(D)_\sigma = \sigma \begin{pmatrix} H_{k_1}^D \\ H_{k_2}^D \\ \vdots \\ H_{k_K}^D \end{pmatrix}, \quad (7.8)$$

em que K é o número máximo de graus da rede. Para descrição, o vetor de características da média $F(D_1)_\mu$ é combinado considerando diferentes valores de D . Assim, é fornecido o vetor de características $\vec{Y}^\mu_{D_1, D_2, \dots, D_n}$ como segue:

$$\vec{Y}^\mu_{D_1, D_2, \dots, D_n} = [F(D_1)_\mu, F(D_2)_\mu, \dots, F(D_n)_\mu]. \quad (7.9)$$

O vetor de características do desvio padrão $[F(D_1)_\sigma]$ também é combinado usando

diferentes valores de D para compor o vetor de características $\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\sigma$, de acordo com

$$\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\sigma = [F(D_1)_\sigma, F(D_2)_\sigma, \dots, F(D_n)_\sigma]. \quad (7.10)$$

Finalmente, um vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$ é proposto baseado na concatenação dos vetores de características $\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^g$, $\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\mu$ e $\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\sigma$. Este vetor de característica final combina diferentes informações sobre os TEPs e, conseqüentemente, sobre a estrutura da rede, melhorando o desempenho de classificação. Nos experimentos, esse vetor de características obteve o melhor resultado de classificação. Assim, o vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$ é dado pela concatenação:

$$\vec{\Psi}_{D_1, D_2, \dots, D_n} = [\vec{\Upsilon}_{D_1, D_2, \dots, D_n}^g, \vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\mu, \vec{\Upsilon}_{D_1, D_2, \dots, D_n}^\sigma]. \quad (7.11)$$

7.1.4 Complexidade Computacional

Considere uma rede complexa com N vértices e grau médio $\langle k \rangle$. Para calcular a complexidade computacional da abordagem LLNA-BP, primeiro analisa-se a complexidade do LLNA, que fornece o TEP. Dada uma regra, a abordagem LLNA calcula o próximo estado de cada vértice com base nos estados dos vértices vizinhos. Portanto, para cada iteração do LLNA, $N \times \langle k \rangle$ operações são necessárias. Uma vez que o LLNA é iterado por T passos de tempo, para obter um TEP, $N \times \langle k \rangle \times T$ operações são necessárias. Portanto, a complexidade computacional da abordagem LLNA para obter um TEP é dada por $O(N \times \langle k \rangle \times T)$.

A abordagem proposta calcula o padrão binário para cada vértice da rede considerando o tamanho do dicionário D . Assim, $N \times T \times D$ operações são realizadas a fim de obter os histogramas dos padrões binários. Portanto, considerando todas as etapas do método LLBP-BP, a complexidade computacional é $O((N \times T) \times (\langle k \rangle + D))$. Como o valor de D é pequeno comparado a T e N (e.g., $D = 7$, $N = 500$, $T = 300$), este valor pode ser suprimido, resultando em uma complexidade de $O((N \times T) \times (\langle k \rangle))$.

7.2 Experimentos

Nesta seção, as estratégias de validação, bases de dados de redes complexas e métodos comparados são detalhados.

7.2.1 Bases de dados

A seguir, as bases de dados usados nos experimentos são descritas.

- **4 Modelos:** essa base de dados é composta por redes sintéticas construídas de acordo com quatro modelos: (i) aleatório (ERDöS; RéNYI, 1959), com probabilidade de conexão

entre dois vértices $p = \langle k \rangle / n$; (ii) pequeno mundo (WATTS; STROGATZ, 1998), com probabilidade de reconexão de $p = 0, 1$; (iii) livre de escala (BARABÁSI; ALBERT, 1999), com conexão preferencial linear e não linear, e, (iv) geográfica (WAXMAN, 1988). Para cada modelo, amostras de redes foram geradas com os seguintes parâmetros : $\langle k \rangle$: 4, 6, 8, 10, 12, 14, 16; e, $N = 500, 1000, 1500$ e 2000. Para cada uma das 28 combinações de $\langle k \rangle - N$ foram geradas 100 redes. Assim, o número de redes para cada modelo é igual a 2800, resultando em uma base de dados com um total de 11200 redes complexas.

- **4 Modelos + $\langle k \rangle$:** o segundo experimento foi executado para classificar as 28 combinações $\langle k \rangle - N$, sendo assim, nesse experimento são 28 classes com 100 amostras cada.
- **Redes com ruído:** outro experimento foi executado para avaliar a robustez da tolerância a ruído. Para tal, a topologia da rede foi modificada através da remoção e adição de arestas de acordo com uma taxa de ruído ρ . A taxa de ruído ρ indica que $\frac{\rho}{2}$ das arestas são adicionadas em relação ao número total de arestas, e, $\frac{\rho}{2}$ são removidas. Portanto, à medida que o ρ aumenta, mais mudanças estruturais são realizadas na topologia da rede (MIRANDA; MACHICAO; BRUNO, 2016). A análise da robustez a ruído foi realizada no conjunto de dados sintéticos de 4 modelos para três diferentes valores de ρ : 0,1, 0,2 e 0,3.
- **Livre de escala:** essa base de dados contém redes livres de escala geradas usando dois modelos diferentes: Barabási & Albert (BARABÁSI; ALBERT, 1999) e Dorogovtsev & Mendes (DOROGOVTSSEV; MENDES, 2002). Para o modelo Barabási & Albert, redes foram geradas com conexões preferenciais lineares e não-lineares (α): 0.5, 1.0, 1.5 e 2.0. Desta forma, este banco de dados tem cinco classes com 100 redes ($N = 1000$ nós e $\langle k \rangle = 8$) por classe.
- **Social:** composto por redes da plataforma SNAP (*Stanford Network Analysis Project*) (LESKOVEC; KREVL, 2014; MCAULEY; LESKOVEC, 2012). Cada rede social (ou *ego-networks*) corresponde às relações sociais ou amigos de um usuário específico que não é representado na rede. Este conjunto de dados contém 100 amostras divididas em duas classes (Google+ e Twitter) com 50 amostras para cada uma.
- **Redes Metabólicas:** essa base de dados é composta de redes metabólicas que foram construídas usando o modelo de rede substrato-produto (ZHAO *et al.*, 2006), assim, metabólitos (vértices) de substratos estão ligados a metabólitos de produtos por cada reação. As reações bioquímicas de vários organismos foram obtidas do banco de dados *Kyoto Encyclopedia of Genes and Genomes* (KEGG) (KANEHISA *et al.*, 2015). Esta base de dados foi subdividida em 7 conjuntos, como segue:
 - *Kingdom*: essa base de dados compreende espécies do domínio da vida *eukaryota*, que consiste em quatro reinos: *animals*, *plantas*, *fungi* e *protist*, cada um deles contendo 40 redes.

- *Animal*: esse conjunto contém quatro classes: *mammals*, *birds*, *peixes* e *insetos*, contendo 14 amostras por classe.
 - *Fungi*: contém quatro classes: *saccharomycetes*, *sordariomycetes*, *eurotiomycetes* e *basidiomycetes*, cada uma delas compostas por 15 redes.
 - *Plant*: contém três classes *Monocots*, *Green algae* e *Eudicots*, compostas por 9 organismos por classe.
 - *Protist*: essa base de dados compreende quatro classes: *Amoebozoa*, *Alveolates*, *Stramenopiles* e *Euglenozoa*, cada classe contém 5 organismos.
 - *Firmicutes-Bacillis*: esse conjunto de dados possui quatro classes *Bacillus*, *Staphylococcus*, *Streptococcus* e *Lactobacillus*, contendo números de espécies desbalanceados, 122, 76, 133 e 83, respectivamente.
 - *Actinobacteria*: essa também é uma base desbalanceada, possuindo três classes: *Mycobacterium*, *Corynebacterium* e *Streptomyces* com 60, 86, e 53 espécies, respectivamente.
- Seleção de regras Redes Metabólicas: como a escolha das melhores regras do LLNA deve ser realizada em um banco de dados exclusivo, sete bancos de dados foram gerados contendo redes metabólicas de outros organismos das mesmas classes, a serem utilizados para a seleção da regra do LLNA. Assim, o conjunto *kingdom-selection* compreende 9 amostras para cada classe. Os conjuntos *animal-selection*, *fungi-selection*, *plant-selection* e *protist-selection* possuem 2 amostras por cada classe, enquanto os conjuntos *firmicutes-bacillis-selection* e *actinobacteria-selection* contém 10 redes por classe.

7.2.2 Configuração do LLNA-BP

Esta seção detalha a configuração do LLNA para gerar os TEPs usados nos experimentos. Como o objetivo da abordagem proposta é melhorar a caracterização dos TEPs, os mesmos parâmetros com relação às especificações do LLNA do trabalho original foram utilizados (MIRANDA; MACHICAO; BRUNO, 2016). Portanto, a configuração inicial s_0 foi setada aleatoriamente usando probabilidade de 50% de células vivas, e os autômatos de rede evoluíram durante $T = 350$ iterações. Além disso, as mesmas regras ótimas selecionadas foram adotadas para cada uma das bases de dados usados por Miranda, Machicao e Bruno (2016). Assim, é possível comparar de forma justa a abordagem proposta e o método anterior.

A regra B135678-S03456 foi usada para a base 4 Modelos e nas redes com ruído. Para a base de redes livre de escala foi considerada a regra B0157-S457, enquanto a regra B01678-S0457 foi usada para a base 4 Modelos + $\langle k \rangle$. Em relação às aplicações reais, para a base Social foi utilizada a regra B0167-S248.

Como neste trabalho foi introduzida a base de dados de redes metabólicas, o procedimento de seleção de regras (consulte a Seção 2.3.1) também foi realizado usando uma base de seleção

de regras de redes Metabólicas, que contém um pequeno número de outras redes metabólicas para cada um dos problemas de classificação. Calculou-se a seleção de regras seguindo os mesmos critérios estabelecidos por [Miranda, Machicao e Bruno \(2016\)](#). Como resultado, as seguintes regras ótimas foram selecionadas: B02345678-S123468, B023468-S01468, B04-S1468, B0468-S0467, B0236-S123567, B0468-S0458, B1237-S267 para *Kingdom*, *Animal*, *Fungi*, *Plant*, *Protist*, *Firmicutes-Bacillis* e *Actibacteria*, respectivamente.

7.2.3 Configuração de Classificação

No processo de classificação, empregou-se a estratégia de validação cruzada com 10 pastas, que é uma forma generalizada de avaliar a capacidade de predição de um modelo. Essa estratégia divide o conjunto de dados em 10 subconjuntos iguais ou quase iguais, um subconjunto é usado para teste e o restante para treinamento. Todos os subconjuntos são usados como conjunto de testes. Como essa estratégia de validação não é determinística, o procedimento de validação cruzada foi aplicado 100 vezes e a acurácia média e o desvio padrão são considerados como o desempenho do método. O classificador SVM foi usado nos experimentos de modo a seguir a configuração estabelecida por [Miranda, Machicao e Bruno \(2016\)](#). Esse classificador usa os hiperplanos como limites de decisão e a classificação é realizada encontrando o hiperplano que produz a separação máxima entre duas classes ([HEARST et al., 1998](#)).

7.2.4 Métodos para Comparação

Neste trabalho, o desempenho do LLNA-BP foi comparado com o LLNA anterior ([MIRANDA; MACHICAO; BRUNO, 2016](#)). Portanto, os resultados obtidos por [Miranda, Machicao e Bruno \(2016\)](#) foram reproduzidos usando a entropia de Shannon, comprimento de palavra e a distribuição do histograma de complexidade de Lempel-Ziv dada por $[\vec{\mu}_S, \vec{\mu}_W, \vec{\mu}_L]$ como o vetor de características. Além da comparação com o método LLNA, o LLNA-BP também foi comparado com as medidas topológicas clássicas de redes. Como sugerido por [Costa et al. \(2010\)](#), um conjunto de medidas de RC de diferentes categorias, tais como, grau médio, distribuições de graus, correlações, distâncias e comprimentos de caminho, hierárquico, medidas espectrais, transitividade, coeficiente de agrupamento, entre outros, são comumente usados para caracterizar a topologia de redes. Portanto, a combinação usada foi: grau médio ($\langle k \rangle$), grau hierárquico médio de nível 1 ($\langle H_{k_1} \rangle$) e nível 2 ($\langle H_{k_2} \rangle$), coeficiente de cluster médio ($\langle cc \rangle$), comprimento médio do caminho (l) e grau de correlação de Pearson (ρ_P) ([COSTA et al., 2007](#); [COSTA et al., 2010](#)). Mais especificamente, as equações são dadas da seguinte maneira:

- $\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i$;
- $\langle H_{k_h} \rangle = \frac{1}{N} \sum_i k_i^h$, quando $h = 2$ representa a soma dos graus da vizinhança dos vértice v_i . Da mesma forma, quando $h = 3$ representa a soma dos graus dos vizinhos dos vizinhos do vértice v_i ;

- $\langle cc \rangle = \frac{1}{N} \sum_i cc_i$, em que o coeficiente de *clustering* é definido por $cc_i = \frac{e_i}{k_i(k_i-1)}$ e e_i é o número de pares conectados para cada um dos vizinhos de v_i ;
- $\langle l \rangle$, é o comprimento médio dos caminhos mais curtos entre quaisquer dois vértices da rede;
- ρ_P é uma medida de assortatividade da rede.

7.3 Resultados e Discussões

Esta seção reporta os resultados experimentais obtidos na tarefa de classificação aplicada a todas as bases de dados citadas anteriormente. O desempenho do método proposto também foi analisado usando diferentes parâmetros. Finalmente, também comparou-se o desempenho de classificação entre o método proposto, medidas tradicionais de rede complexa e a abordagem anterior LLNA.

7.3.1 Avaliação dos Parâmetros

Para aplicar eficientemente a abordagem proposta na classificação de redes, primeiro deve-se definir o vetor de características ideal e o comprimento do padrão binário D . A [Tabela 23](#) mostra os resultados obtidos na base de dados de rede Social ao usar valores diferentes do comprimento do padrão binário D para calcular os vetores de características propostos $\vec{Y}_{D_1, D_2, \dots, D_n}^g$, $\vec{Y}_{D_1, D_2, \dots, D_n}^\mu$, $\vec{Y}_{D_1, D_2, \dots, D_n}^\sigma$ e $\vec{\Psi}_{D_1, D_2, \dots, D_n}$. Os resultados referem-se a porcentagem de amostras de redes classificadas corretamente usando cada parâmetro. Esses resultados mostram que o método proposto é discriminativo, alcançando bons resultados para todos os vetores de características e combinações de D . Com relação aos diferentes vetores de características, os melhores resultados foram obtidos com o vetor $\vec{\Psi}_{D_1, D_2, \dots, D_n}$. Esse vetor de características contém mais informações sobre os padrões de rede, pois combina características globais e específicas de vértices com grau semelhante.

Considerando apenas o vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$, as [Tabelas 24](#) e [25](#) mostram os resultados para bases de dados sintéticas e metabólicas, respectivamente, usando diferentes valores e combinações de D . Nas bases de dados *Animal* e *Fungi* pode-se notar que a acurácia tende a aumentar à medida que o valor de D é incrementado. Por outro lado, os resultados obtidos nas bases de dados sintética, Social e *Kingdom* mostram uma alta taxa de classificação correta para qualquer combinação de valores de D . Dessa forma, para esses testes, não foi combinado mais de três valores de D para compor os vetores de características devido ao grande número de descritores. Além disso, os resultados mostram uma pequena melhoria na taxa de classificação quando é combinado mais de dois valores de D . Portanto, é sugerido o vetor de características $\vec{\Psi}_{5,7}$ para ser usado como um parâmetro padrão da abordagem, pois representa um bom equilíbrio entre o desempenho e o número de descritores em todas as bases de dados.

Tabela 23 – Resultados para a base de dados de rede social usando diferentes vetores de características e combinações de valores de D .

$\{D_1, D_2, \dots, D_n\}$	$\vec{\Upsilon}^g_{D_1, D_2, \dots, D_n}$	$\vec{\Upsilon}^\mu_{D_1, D_2, \dots, D_n}$	$\vec{\Upsilon}^\sigma_{D_1, D_2, \dots, D_n}$	$\vec{\Psi}_{D_1, D_2, \dots, D_n}$
{4}	93,00 ($\pm 0,00$)	93,00 ($\pm 0,00$)	91,00 ($\pm 0,00$)	92,80 ($\pm 0,40$)
{5}	92,70 ($\pm 0,64$)	93,00 ($\pm 0,00$)	90,90 ($\pm 0,30$)	92,40 ($\pm 0,49$)
{6}	93,10 ($\pm 0,30$)	93,20 ($\pm 0,75$)	90,20 ($\pm 0,40$)	92,70 ($\pm 0,64$)
{7}	93,00 ($\pm 0,00$)	93,20 ($\pm 0,40$)	90,20 ($\pm 0,60$)	92,50 ($\pm 1,02$)
{8}	92,90 ($\pm 0,70$)	92,00 ($\pm 0,63$)	89,30 ($\pm 0,90$)	92,50 ($\pm 1,02$)
{4, 5}	93,00 ($\pm 0,63$)	93,00 ($\pm 0,00$)	90,90 ($\pm 0,30$)	92,30 ($\pm 0,46$)
{4, 6}	93,00 ($\pm 0,00$)	93,10 ($\pm 0,70$)	90,10 ($\pm 0,54$)	92,40 ($\pm 0,66$)
{4, 7}	93,00 ($\pm 0,00$)	93,10 ($\pm 0,30$)	90,30 ($\pm 0,46$)	92,30 ($\pm 0,78$)
{4, 8}	92,80 ($\pm 0,87$)	91,70 ($\pm 0,78$)	89,30 ($\pm 1,19$)	92,80 ($\pm 0,75$)
{5, 6}	93,20 ($\pm 0,40$)	93,10 ($\pm 0,54$)	90,00 ($\pm 0,45$)	92,50 ($\pm 0,67$)
{5, 7}	93,10 ($\pm 0,30$)	93,10 ($\pm 0,30$)	90,00 ($\pm 0,63$)	93,40 ($\pm 0,92$)
{5, 8}	92,80 ($\pm 0,87$)	91,70 ($\pm 0,78$)	89,40 ($\pm 1,02$)	92,90 ($\pm 0,70$)
{6, 7}	93,10 ($\pm 0,30$)	93,00 ($\pm 0,45$)	89,90 ($\pm 0,54$)	93,40 ($\pm 1,02$)
{6, 8}	92,80 ($\pm 0,87$)	91,60 ($\pm 0,66$)	89,80 ($\pm 0,87$)	92,90 ($\pm 1,04$)
{7, 8}	92,00 ($\pm 0,63$)	91,40 ($\pm 0,66$)	89,20 ($\pm 1,08$)	92,70 ($\pm 1,00$)
{4, 5, 6}	93,20 ($\pm 0,40$)	93,10 ($\pm 0,30$)	89,90 ($\pm 0,54$)	92,60 ($\pm 0,49$)
{4, 5, 7}	93,10 ($\pm 0,30$)	93,00 ($\pm 0,45$)	90,10 ($\pm 0,70$)	93,60 ($\pm 0,92$)
{4, 5, 8}	92,80 ($\pm 0,87$)	91,70 ($\pm 0,78$)	89,40 ($\pm 1,02$)	93,10 ($\pm 0,70$)
{4, 6, 7}	93,10 ($\pm 0,30$)	93,00 ($\pm 0,45$)	90,00 ($\pm 0,45$)	93,70 ($\pm 0,78$)
{4, 6, 8}	92,70 ($\pm 0,90$)	91,60 ($\pm 0,66$)	89,40 ($\pm 1,02$)	93,20 ($\pm 0,87$)
{4, 7, 8}	92,00 ($\pm 0,63$)	91,40 ($\pm 0,66$)	89,10 ($\pm 0,83$)	92,60 ($\pm 1,02$)
{5, 6, 7}	93,10 ($\pm 0,30$)	92,90 ($\pm 0,54$)	89,90 ($\pm 0,54$)	93,90 ($\pm 0,83$)
{5, 6, 8}	92,60 ($\pm 0,80$)	91,60 ($\pm 0,66$)	89,60 ($\pm 0,80$)	93,30 ($\pm 0,78$)
{5, 7, 8}	91,90 ($\pm 0,70$)	91,40 ($\pm 0,66$)	89,10 ($\pm 0,83$)	93,00 ($\pm 1,18$)
{6, 7, 8}	91,70 ($\pm 0,78$)	91,30 ($\pm 0,64$)	88,70 ($\pm 0,78$)	93,20 ($\pm 1,08$)

Fonte: Ribas, Machicao e Bruno (2020).

Esse vetor de características possui 480 descritores, que é menor em comparação com o tamanho de outros vetores propostos, como o $\vec{\Psi}_{6,7,8}$, que possui 1344 atributos.

7.3.2 Comparações e Discussões

Para expandir a avaliação da abordagem proposta, os seus resultados foram comparados com métricas tradicionais de redes complexas (grau hierárquico médio do nível 2 e 3, coeficiente de agrupamento médio, caminho mínimo médio e coeficiente de Person) e o LLNA tradicional (MIRANDA; MACHICAO; BRUNO, 2016). As medidas de rede clássicas usadas neste experimento são descritas na Subseção 7.2.4. Para o LLNA tradicional, os resultados obtidos por Miranda, Machicao e Bruno (2016) foram replicados. Para a abordagem proposta, o vetor de características $\vec{\Psi}_{5,7}$ foi considerado em todas as bases de dados, uma vez que este atingiu um bom desempenho, como visto na Subseção 7.3.1.

Tabela 24 – Resultados para diferentes combinações de valores de D com o vetor de características $\vec{\Psi}_{D_1, D_2, \dots, D_n}$ em bases de dados sintéticas. Desvios padrão entre $\pm 0,0$ e $\pm 0,09$ não são mostrados.

$\{D_1, D_2, \dots, D_n\}$	4 Modelos	4 Modelos + $\langle k \rangle$	Livre de Escala	Redes com ruídos		
				$\rho = 0,1$	$\rho = 0,2$	$\rho = 0,3$
{4}	99,53	94,39	99,36 ($\pm 0,23$)	99,44	99,09	98,65
{5}	99,98	96,71	99,46 ($\pm 0,21$)	99,96	99,94	99,77
{6}	100,00	97,60	99,54 ($\pm 0,19$)	100,00	99,99	99,97
{7}	100,00	98,23	99,58 ($\pm 0,22$)	100,00	99,98	99,99
{8}	100,00	98,58	99,70 ($\pm 0,19$)	100,00	99,99	99,97
{4, 5}	99,98	96,99	99,32 ($\pm 0,21$)	99,95	99,94	99,75
{4, 6}	100,00	97,75	99,32 ($\pm 0,25$)	100,00	99,99	99,96
{4, 7}	100,00	98,27	99,36 ($\pm 0,23$)	100,00	99,99	99,99
{4, 8}	100,00	98,59	99,38 ($\pm 0,26$)	100,00	99,99	99,97
{5, 6}	100,00	97,90	99,50 ($\pm 0,22$)	100,00	99,99	99,96
{5, 7}	100,00	98,31	99,52 ($\pm 0,19$)	100,00	99,98	99,99
{5, 8}	100,00	98,61	99,52 ($\pm 0,19$)	100,00	99,99	99,97
{6, 7}	100,00	98,40	99,56 ($\pm 0,21$)	100,00	99,99	99,99
{6, 8}	100,00	98,63	99,56 ($\pm 0,21$)	100,00	99,99	99,98
{7, 8}	100,00	98,68	99,66 ($\pm 0,25$)	100,00	99,99	99,98
{4, 5, 6}	100,00	97,96	99,46 ($\pm 0,27$)	100,00	99,99	99,96
{4, 5, 7}	100,00	98,39	99,48 ($\pm 0,19$)	100,00	99,98	99,99
{4, 5, 8}	100,00	98,65	99,46 ($\pm 0,25$)	100,00	99,99	99,97
{4, 6, 7}	100,00	98,46	99,42 ($\pm 0,24$)	100,00	99,98	99,98
{4, 6, 8}	100,00	98,66	99,42 ($\pm 0,24$)	100,00	99,99	99,97
{4, 7, 8}	100,00	98,70	99,46 ($\pm 0,19$)	100,00	99,99	99,98
{5, 6, 7}	100,00	98,48	99,54 ($\pm 0,19$)	100,00	99,98	99,99
{5, 6, 8}	100,00	98,68	99,52 ($\pm 0,19$)	100,00	99,99	99,98
{5, 7, 8}	100,00	98,72	99,52 ($\pm 0,19$)	100,00	99,99	99,98
{6, 7, 8}	100,00	98,73	99,58 ($\pm 0,22$)	100,00	99,99	99,98

Fonte: Ribas, Machicao e Bruno (2020).

A Tabela 26 apresenta os resultados obtidos para cada abordagem avaliada em diferentes bases de dados de rede. Os resultados mostraram que para as bases 4 Modelos e a de ruído o método proposto e outros obtiveram precisões muito semelhantes. Por outro lado, em relação ao conjunto de dados 4 Modelos + $\langle k \rangle$, o LLNA-BP (98,31%) superou o LLNA tradicional (90,76%) e as métricas clássicas de rede (65,20%). Além disso, na base de dados livre de escala, o método proposto apresenta uma pequena melhora de 1,22% e 3,32% em comparação com o LLNA tradicional e as medidas clássicas de rede, respectivamente. As altas acurácias nas bases de dados sintéticas são esperadas, visto que estas bases possuem propriedades intrínsecas geradas por modelos matemáticos previamente conhecidos. Assim, essas redes são utilizadas como um estudo inicial para verificar a viabilidade e capacidade discriminativa das abordagens.

O método proposto também foi comparado em outras bases de dados de redes reais (veja a Tabela 26). A tarefa de classificação dessas bases de dados é um problema desafiador, pois cada aplicação tem propriedades diferentes, permitindo, assim, avaliar a capacidade de generalização

Tabela 25 – Acurácia (%) e desvio padrão para a classificação das três base de dados de metabólitos usando diferentes combinações de valores de D com o vetor de características $\tilde{\Psi}_{D_1, D_2, \dots, D_n}$.

$\{D_1, D_2, \dots, D_n\}$	Kingdom	Animal	Fungi
{4}	94,12 ($\pm$ 6,08)	68,03 ($\pm$ 20,30)	62,00 ($\pm$ 17,90)
{5}	96,19 ($\pm$ 4,69)	78,00 ($\pm$ 16,65)	71,50 ($\pm$ 17,62)
{6}	97,06 ($\pm$ 4,57)	78,43 ($\pm$ 16,31)	75,00 ($\pm$ 17,33)
{7}	97,25 ($\pm$ 4,20)	83,63 ($\pm$ 15,52)	77,17 ($\pm$ 16,69)
{8}	96,56 ($\pm$ 4,20)	81,53 ($\pm$ 16,39)	77,50 ($\pm$ 17,14)
{4, 5}	96,19 ($\pm$ 4,69)	74,80 ($\pm$ 17,36)	70,67 ($\pm$ 15,91)
{4, 6}	97,12 ($\pm$ 4,21)	78,53 ($\pm$ 17,35)	74,67 ($\pm$ 17,80)
{4, 7}	97,50 ($\pm$ 4,07)	82,57 ($\pm$ 16,02)	77,67 ($\pm$ 17,12)
{4, 8}	96,56 ($\pm$ 4,20)	81,77 ($\pm$ 15,97)	78,83 ($\pm$ 16,56)
{5, 6}	96,69 ($\pm$ 4,65)	78,03 ($\pm$ 17,29)	74,50 ($\pm$ 17,32)
{5, 7}	97,44 ($\pm$ 3,98)	84,87 ($\pm$ 15,25)	76,17 ($\pm$ 17,45)
{5, 8}	96,69 ($\pm$ 4,11)	82,27 ($\pm$ 15,75)	78,33 ($\pm$ 16,33)
{6, 7}	97,19 ($\pm$ 4,29)	82,93 ($\pm$ 14,61)	76,67 ($\pm$ 17,08)
{6, 8}	96,62 ($\pm$ 4,30)	82,47 ($\pm$ 15,59)	78,17 ($\pm$ 16,87)
{7, 8}	96,56 ($\pm$ 4,20)	82,60 ($\pm$ 15,33)	77,50 ($\pm$ 16,65)
{4, 5, 6}	96,69 ($\pm$ 4,65)	77,60 ($\pm$ 17,44)	73,83 ($\pm$ 17,29)
{4, 5, 7}	97,50 ($\pm$ 3,87)	82,73 ($\pm$ 15,84)	76,33 ($\pm$ 16,95)
{4, 5, 8}	96,75 ($\pm$ 4,12)	81,20 ($\pm$ 15,98)	79,17 ($\pm$ 16,30)
{4, 6, 7}	97,44 ($\pm$ 4,08)	83,10 ($\pm$ 15,51)	76,50 ($\pm$ 18,06)
{4, 6, 8}	96,81 ($\pm$ 4,21)	82,27 ($\pm$ 15,57)	78,83 ($\pm$ 16,73)
{4, 7, 8}	96,62 ($\pm$ 4,11)	82,60 ($\pm$ 15,85)	78,33 ($\pm$ 16,50)
{5, 6, 7}	97,44 ($\pm$ 3,98)	82,80 ($\pm$ 15,16)	75,67 ($\pm$ 18,11)
{5, 6, 8}	96,94 ($\pm$ 4,12)	82,63 ($\pm$ 15,41)	78,50 ($\pm$ 16,29)
{5, 7, 8}	96,75 ($\pm$ 4,12)	82,77 ($\pm$ 16,12)	77,83 ($\pm$ 16,42)
{6, 7, 8}	96,69 ($\pm$ 4,30)	83,37 ($\pm$ 15,87)	78,50 ($\pm$ 16,46)

Fonte: Ribas, Machicao e Bruno (2020).

do método. Quanto à tarefa de identificar padrões estruturais nas redes sociais, o método proposto obteve a maior taxa de acurácia quando comparado às demais abordagens. O método proposto forneceu uma melhoria de acurácia de 1,4% em comparação ao LLNA tradicional, e de 5,4% para as medidas clássicas, demonstrando ser um método com um bom poder discriminativo.

Em relação à tarefa de identificar organismos utilizando redes metabólicas, o método proposto também apresentou maior acurácia que os demais. Na base de dados *Fungi*, os resultados experimentais indicam que o método LLNA-BP melhora significativamente a acurácia, por exemplo, de 54,58 % ($\pm$ 19,38) para 76,17% ($\pm$ 17,45) sobre o LLNA tradicional e de 54,90% ($\pm$ 15,39) para 76,17% ($\pm$ 17,45) sobre as medidas clássicas de rede. Na base de dados *Animals*, uma acurácia de 84,87% ($\pm$ 15,25) é alcançada pelo método proposto, que é seguido por uma acurácia de 83,71% ($\pm$ 15,29) e 77,25% ($\pm$ 16,29) obtidos pelas medidas clássicas de rede e pelo LLNA tradicional, respectivamente. Para a base de dados *Kingdom*, a acurácia máxima obtida com o método proposto foi de 97,44% ($\pm$ 3,98) em contraste com 96,61% ($\pm$ 4,33) usando as

medidas clássicas de rede como um vetor de características. Nos resultados, os altos valores de desvio padrão podem ser explicados pela dificuldade dos problemas e pelo número reduzido de amostras usadas para treinar o modelo.

No experimento utilizando a base *Plant*, o método proposto atingiu a maior acurácia, 74,81% ($\pm 5,64$). Na base *Protist*, o método proposto LLNA-BP superou o LLNA anterior e as medidas estruturais em 18,45% e 41,90%, respectivamente. Nas outras duas bases de dados (*Firmicutes-Bacillis* e *Actinobacteria*), o LLNA-BP também obteve melhor desempenho em comparação aos demais. Note que em algumas bases de dados é evidente que as medidas estruturais não conseguem lidar com a complexidade da topologia das redes e, portanto, não são uma boa opção para caracterizá-las. De fato, muitas redes complexas com uma mesma distribuição de grau possuem características específicas das suas estruturas que a diferenciam. Além disso, os resultados demonstram que a assinatura de padrão binário pode caracterizar melhor a complexidade dos TEPs gerados pelo LLNA, aumentando a acurácia em todas as bases de dados quando comparado ao LLNA tradicional.

Tabela 26 – Comparação da acurácia (%) e desvio padrão obtidos pela abordagem proposta, o LLNA tradicional e as medidas de rede aplicadas na classificação de 14 bases de dados de rede. O LLNA-BP usou o vetor de características $\vec{\Psi}_{5,7}$. O LLNA usou $[\vec{\mu}_S, \vec{\mu}_W, \vec{\mu}_L]$ como vetor de características, enquanto o vetor das medidas clássicas de rede foi $[\langle k \rangle, \langle H_{k_1} \rangle, \langle H_{k_2} \rangle, \langle cc \rangle, l, \rho_P]$.

Bases de dados		Abordagens		
		LLNA-BP	LLNA	Medidas clássicas
Sintéticas	<i>4 Modelos</i>	100,0 ($\pm 0,00$)	99,992 ($\pm 0,002$)	100,0 ($\pm 0,00$)
	<i>4 Modelos + $\langle k \rangle$</i>	98,31 ($\pm 0,02$)	90,76 ($\pm 0,07$)	65,20 ($\pm 0,20$)
	<i>Livre de Escala</i>	99,52 ($\pm 0,19$)	98,30 ($\pm 0,2$)	96,20 ($\pm 0,04$)
	<i>Ruído $\rho = 10\%$</i>	100,0 ($\pm 0,00$)	99,983 ($\pm 0,004$)	100,0 ($\pm 0,00$)
	<i>Ruído $\rho = 20\%$</i>	99,98 ($\pm 0,00$)	99,97 ($\pm 0,01$)	100,0 ($\pm 0,00$)
	<i>Ruído $\rho = 30\%$</i>	99,99 ($\pm 0,00$)	99,95 ($\pm 0,01$)	100,0 ($\pm 0,00$)
Reais	<i>Social</i>	93,40 ($\pm 0,92$)	92,00 ($\pm 1,00$)	88,00 ($\pm 2,00$)
	<i>Kingdom</i>	97,44 ($\pm 3,98$)	93,10 ($\pm 5,38$)	96,61 ($\pm 4,33$)
	<i>Animal</i>	84,87 ($\pm 15,25$)	77,25 ($\pm 16,29$)	83,71 ($\pm 15,29$)
	<i>Fungi</i>	76,17 ($\pm 17,45$)	54,58 ($\pm 19,38$)	54,90 ($\pm 15,39$)
	<i>Plant</i>	74,81 ($\pm 5,64$)	69,70 ($\pm 4,67$)	54,19 ($\pm 9,17$)
	<i>Protist</i>	87,00 ($\pm 5,29$)	68,55 ($\pm 6,30$)	45,10 ($\pm 10,02$)
	<i>Firmicutes-Bacillis</i>	98,30 ($\pm 1,17$)	84,63 ($\pm 2,00$)	95,67 ($\pm 0,59$)
	<i>Actinobacteria</i>	95,13 ($\pm 1,22$)	91,48 ($\pm 1,60$)	93,16 ($\pm 0,70$)

Fonte: Ribas, Machicao e Bruno (2020).

7.3.3 Tempo de Processamento

Para avaliar o tempo de processamento, redes da base de dados sintética livre de escala foram consideradas com $\langle k \rangle = 16$. O método proposto LLNA-BP levou em média 0,10s, 0,21s, 0,32s e 0,45s para calcular as assinaturas das redes com $N = 500, 1000, 1500$ e 2000 vértices,

respectivamente. Por outro lado, o método LLNA tradicional levou em média 0,04s, 0,09s, 0,14s e 0,19s para obter as assinaturas das mesmas amostras de rede, respectivamente. Para esses testes utilizou-se um processador Intel(R) Core i7 de 3,60 GHz, 64 GB de RAM e sistema operacional de 64 bits. Embora o LLNA-BP tenha gasto um tempo de execução ligeiramente maior em comparação com o LLNA tradicional, os resultados indicam que ambas as abordagens têm tempos de execução competitivos para aplicações em tempo real.

7.4 Considerações Finais

Este capítulo apresentou uma maneira inovadora de melhorar a extração de características de padrões de evolução no tempo gerados pela técnica LLNA no contexto de reconhecimento de padrões. Assim, para representar suas sequências de zeros e uns foi proposto usar frequências de padrões binários, a partir dos quais vários descritores foram usados para caracterização da rede. A robustez do método proposto foi demonstrada com respeito a dois tipos de bases de dados: redes sintéticas e redes do mundo real. Primeiramente, o efeito dos parâmetros envolvidos foi investigado no desempenho de classificação nas bases de dados. A partir desse estudo, definiu-se um conjunto de parâmetros padrão para todos os conjuntos de dados que fornece um bom equilíbrio entre desempenho e número de atributos. Em seguida, os resultados experimentais confirmaram a eficácia do método proposto em relação ao LLNA original e às medidas estruturais obtidas diretamente da topologia da rede complexa. Em particular, o método proposto melhora, significativamente, a precisão em comparação com o LLNA original em aplicações do mundo real. Isso demonstra que mesmo em tarefas desafiadoras, o método proposto leva a uma interpretação eficaz dos padrões complexos presentes nos TEPs. Assim, este estudo indica que investigações mais aprofundadas podem ser realizadas para caracterizar a complexidade dos TEPs, usando também outras estratégias de caracterização como redes neurais artificiais. Para trabalhos futuros, acredita-se que formas otimizadas de encontrar as regras de transição ideais para a base de dados, pode permitir a aplicação rápida do método em diversos problemas. Além disso, a técnica também pode ser estendida para caracterizar redes complexas de imagens. Por fim, além de ser um caso interessante em que os descritores de padrões binários melhoram o desempenho, certamente, esta mesma proposta pode ser estendida para outros TEPs de autômatos celulares.

APLICAÇÕES

Neste capítulo, as abordagens desenvolvidas neste trabalho são aplicadas em diversos problemas reais. O objetivo destas aplicações é verificar o desempenho dos métodos em termos práticos, avaliar as contribuições científicas do trabalho para com outras áreas da ciência e também na solução de problemas relevantes da sociedade. Portanto, a seguir cinco aplicações que envolvem texturas em níveis de cinza, coloridas e também formas são apresentadas e discutidas.

8.1 Reconhecimento de Espécies de Peixes baseado no Contorno do Otólito

Sistemas automáticos de reconhecimento de espécies de peixes têm sido desenvolvidos por pesquisadores de várias áreas, tais como, aprendizado de máquina, análise de imagens e reconhecimento de padrões. A identificação de espécies de peixes pode envolver a análise de várias características morfológicas, incluindo o contorno do otólito (SALIMI *et al.*, 2016). Os otólitos são estruturas localizados no ouvido interno dos peixes (labirinto) formadas por cristais de carbonato de cálcio (BREMM; SCHULZ, 2014). A classificação baseada no otólito é relevante devido à sua composição que é resistente ao trato digestivo dos peixes, permitindo que pesquisadores identifiquem espécies de peixes em materiais estomacais (BREMM; SCHULZ, 2014). Assim, o interesse nessa classificação se dá, principalmente, pela sua importância em ciências paleontológicas e ecológicas.

Esta seção aplica os métodos de análise de formas para a classificação de espécies de peixes baseada no contorno do otólito. Essa tarefa é difícil devido à similaridade nas características morfológicas na forma do otólito das diferentes espécies, sendo elas diferenciadas por detalhes. Nas seções abaixo, a base de dados contendo as imagens de otólito de diferentes espécies e famílias de peixes é descrita e, em seguida, os resultados experimentais utilizando os métodos de análise de formas são discutidos. Os resultados demonstram que os métodos propostos podem ser

uma boa ferramenta para caracterização do otólito, pois atingem altas acurácias nos diferentes experimentos realizados.

8.1.1 Dados

A base de dados, disponibilizada por [Salimi et al. \(2016\)](#), contém imagens dos otólitos sagitais direitos capturadas com um estereomicroscópio (Olympus DP25FW, aumento de 6,3X) e uma câmera digital. Essas imagens foram obtidas com as formas tiradas com a borda dorsal voltada para cima e a extremidade posterior voltada para a direção positiva ([SALIMI et al., 2016](#)). A base contém 14 espécies de peixes de três famílias *Engraulidae* (3), *Sciaenidae* (5) e *Ariidae* (6), em que o número de espécies por família é dado em parenteses. As classes possuem de 28 a 60 amostras cada. A [Figura 51](#) mostra as espécies da família *Ariidae*, com cada linha representando uma espécie e as colunas diferentes espécimes. Para aplicar os métodos de análise de forma, o contorno externo do otólito foi extraído de cada imagem, gerando assim uma imagem binária.

8.1.2 Experimentos e Resultados

Nos experimentos, os métodos propostos e da literatura de análise de forma discutidos no [Capítulo 8](#) são empregados para caracterizar o contorno das formas de otólitos. Para classificar os vetores de características, os algoritmos LDA e SVM são empregados em um esquema de validação *hold-out* específico para cada experimento, conforme definido originalmente na descrição da base em ([SALIMI et al., 2016](#)). Os métodos propostos utilizaram os parâmetros definidos nos experimentos do [Capítulo 6](#): parâmetro padrão para o DTN e os parâmetros da base ETH-80 para o CRSA. Todos os resultados são apresentados na [Tabela 27](#).

Tabela 27 – Resultados comparativos para a classificação de espécies de peixe através do contorno do otólito usando os classificadores LDA e SVM. Quatro experimentos foram realizados, sendo testes com cada família separada e também com todas as famílias juntas.

Métodos	<i>Engraulidae</i>		<i>Sciaenidae</i>		<i>Ariidae</i>		Todas Famílias	
	LDA	SVM	LDA	SVM	LDA	SVM	LDA	SVM
Fourier	96,67	100,0	74,00	74,00	75,00	71,67	73,57	73,57
Curvatura	96,67	96,67	96,00	94,00	83,33	73,33	87,86	83,57
Zernike	93,33	93,33	68,00	76,00	46,67	51,67	56,43	61,43
M. S. Fractal	60,00	73,33	74,00	70,00	48,33	61,67	62,14	55,71
Análise de Segmentos	100,0	96,67	94,00	96,00	85,00	68,33	91,43	80,00
ADCN	100,0	100,0	98,00	98,00	86,67	81,67	95,71	87,86
CND	100,0	96,67	96,00	96,00	91,67	78,33	87,86	86,43
Métodos Propostos								
DTN	96,67	100,0	98,00	100,0	90,00	85,00	90,71	90,71
CRSA	100,0	100,0	98,00	98,00	80,00	86,67	88,57	92,86

Fonte: Elaborada pelo autor.

Figura 51 – Exemplos de imagens da família *Ariidae*. Cada linha representa uma espécie e as colunas são amostras da mesma espécie.

Fonte: [Salimi et al. \(2016\)](#).

O primeiro experimento realizado visa classificar as três espécies da família *Engraulidae*: *Coilia dussumieri*, *Setipinna taty* e *Thryssa hamiltonii*. Nesse experimento, 20 amostras de cada classe são utilizadas para treinar o classificador, enquanto 30 amostras são empregadas no teste (10 amostras por classe). Como pode ser observado na [Tabela 27](#), vários métodos conseguem classificar corretamente todas as amostras, inclusive os métodos propostos DTN e CRSA. No segundo experimento, os espécimes da família *Sciaenidae* são classificados em 5 espécies diferentes. Para tanto, 19 e 10 amostras por classe foram utilizadas para treinamento e teste, respectivamente. Os resultados para essa tarefa também são encorajadores, com o método DTN alcançando a maior acurácia (100%), enquanto outros métodos atingiram 98% ou menos.

Outro experimento considerado é classificar as amostras da família *Ariidae* de acordo

com 6 espécies diferentes. Para essa tarefa, 108 amostras foram utilizadas para treinamento (18 por classe) e outras 60 amostras (10 por classe) são empregadas para testar o modelo de predição. Essa é uma classificação mais difícil devido à similaridade das características morfológicas entre as espécies da família, como pode ser visto nos exemplos da [Figura 51](#). Assim, os melhores resultados foram obtidos pelos métodos CND (91,67%) e DTN (90,00%), seguido pelo CRSA (86,67%) e ADCN (86,67%).

Por fim, um experimento considera os espécimes das três famílias estudadas, resultando em 14 espécies com 18 amostras para treino e 10 para teste cada. Nesse experimento, o método ADCN obteve a melhor acurácia seguida do método proposto CRSA, enquanto o método DTN também alcançou acurácias competitivas. A maior dificuldade do método CRSA foi distinguir as espécies *Cryptarius truncatus* (classe 3) e *Nemapteryx caelata* (classe 8), ambas da família *Ariidae*, que é a mais desafiadora. Esse problema pode ser observado na matriz de confusão para o método CRSA apresentada na [Figura 52](#). Como pode-se perceber na matriz, o espécime da classe 3 em (a) classificado incorretamente como da classe 8, é bastante similar ao exemplo da classe 8 em (b) que foi utilizado no treinamento. Assim, para superar essa limitação e aumentar o desempenho, acredita-se que um ajuste nos parâmetros possa aprimorar a descrição das características específicas do problema, embora ressalta-se que os valores padrão, definidos no estudo de parâmetros, entregam um bom desempenho em diferentes tarefas. Assim, considerando as quatro tarefas de classificação estudadas, os métodos propostos demonstraram ser uma boa opção para o problema de classificação de espécies de peixe baseado na morfologia da forma do otólito.

8.2 Análise de Plasticidade Foliar de Jacaranda Caroba como biossensor natural de poluentes

Jacaranda Caroba é uma espécie de planta arbórea popular no Brasil que está presente em vários biomas, principalmente na região de cerrado. Uma vez que a estrutura foliar da planta pode refletir as condições do ambiente em que ela está, alterações nos padrões de plasticidade podem indicar, por exemplo, a qualidade do ar de um determinado lugar. Assim, neste estudo, a ideia é investigar a possibilidade de usar essa espécie como um biossensor natural de poluentes, identificando a qualidade do ar do ambiente em que está localizado a planta e fenômenos como chuvas ácidas em florestas longe de regiões industriais. Para tanto, pretende-se analisar a plasticidade foliar de imagens hiperespectrais capturadas por microscopia das folhas da planta.

Esta seção discute a aplicação dos métodos de análise de textura em níveis de cinza e colorida, desenvolvidas no [Capítulo 5](#), para caracterizar a estrutura foliar das plantas e classificá-las conforme as condições em que foram expostas. Isso pode facilitar e baratear o controle de poluentes no meio ambiente através de técnicas computacionais. Este estudo foi realizado em colaboração com o doutorando Leonardo Scabini e Adolfo Grosso e com a equipe de pesquisa-

Figura 52 – Matriz de confusão para o resultado do método CRSA usando o classificador SVM (92,86%) no experimento com todas as famílias. (a) Amostra da espécie *Cryptarius truncatus* classificada incorretamente como da espécie *Nemapteryx caelata*. (b) Exemplo de espécime da classe *Nemapteryx caelata* usada no treinamento.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	8	0	1	0	0	0	0	2	0	0	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	5	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	10	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0	0	0	0	0
8	1	0	4	0	0	0	0	8	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	10	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	9	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	10	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	10	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Fonte: Elaborada pelo autor.

doras da Profa. Rosana Kolb, da UNESP de Assis A seguir, os dados utilizados, experimentos e resultados são apresentados.

8.2.1 Dados

Os dados deste estudo, disponibilizados por Falvo (2016), são imagens hiperespectrais coletados usando microscopia de varredura a laser confocal (Zeiss LSM 780), que emprega uma técnica de microscopia de fluorescência. Essa técnica gera imagens com 32 bandas espectrais (canais), pixels de 16 bits, e dimensão de 2048x2048x32. Neste trabalho, para empregar os métodos, as imagens são convertidas em RGB de 8 bits por meio da média espectral. Para o experimento proposto, os espécimes são divididos em seis grupos: expostos à pulverização de fluoreto de potássio (KF) por 60 dias em cinco concentrações 5mg/L, 15mg/L, 30mg/L, 60mg/L e 1162mg/L e o grupo de controle sem exposição a nenhuma substância. Esses espécimes foram semeados e cultivados em ambiente controlado. As imagens são coletadas após o período de exposição de 4 a 5 folhas de 6 amostras de cada grupo. Assim, a base de dados consiste em 148 imagens divididas em 25 amostras de casos de controle e 25, 26, 24, 24 e 24 para os grupos expostos a 5mg/L, 15mg/L, 30mg/L, 60mg/L e 1162mg/L, respectivamente (FALVO, 2016).

Amostras de cada grupo são apresentadas na [Figura 53](#).

Figura 53 – Exemplos de imagens de Jacaranda Caroba. Da esquerda para direita e de cima para baixo as amostras são de 5mg/L, 15mg/L, 30mg/L, 60mg/L, 1162mg/L e Controle.

Fonte: [Falvo \(2016\)](#).

8.2.2 Experimentos e Resultados

Para analisar a plasticidade foliar dos espécimes de Jacaranda Caroba, métodos propostos (usando os seus parâmetros padrão) e da literatura para texturas coloridas e em níveis de cinza foram aplicados nas imagens. Para os métodos de análise de textura em cinza, as imagens foram convertidas em tons de cinza, enquanto para os métodos de textura colorida, as imagens são transformadas em RGB. Para comparar os métodos propostos com as CNNs, também foram realizados experimentos com as imagens redimensionadas para 224x224 pixels. Os vetores de características calculado pelos métodos são classificados empregando o algoritmo LDA em dois

esquemas de validação: *leave-one-out* e 10 pastas com 10 repetições. O objetivo é classificar as amostras nos 6 grupos gerados nos experimentos. Assim, é possível avaliar capacidade de plantas de *Jacaranda Caroba* serem utilizadas como um biossensor natural de poluentes.

Para analisar as imagens, as três técnicas desenvolvidas para caracterização de texturas são aplicadas: ERCNN, PLCNN e SSRNN. Como os métodos ERCNN e PLCNN são para imagens em níveis de cinza, eles foram aplicados de dois modos: (i) na imagem convertida em nível de cinza; e (ii) integrativo, isto é, nos três canais de cores RGB independentemente para, em seguida, concatenar os três vetores de características para formar a representação final. Os métodos clássicos em níveis de cinza considerados foram: LPQ (OJANSIVU; HEIKKILÄ, 2008), GLDM (WESZKA; DYER; ROSENFELD, 1976), AHP (ZHU *et al.*, 2015), LBP (OJALA; PIETIKAINEN; MAENPAA, 2002) e Fractal (BACKES; CASANOVA; BRUNO, 2009b). Além desses, os modelos de CNNs InceptionV3 (SZEGEDY *et al.*, 2016) e ResNet50 (HE *et al.*, 2016) foram empregados neste estudo em três diferentes abordagens: (i) extração de características aplicando o GAP na última camada convolucional para gerar um vetor unidimensional (LIN; CHEN; YAN, 2013); (ii) treinando o modelo do zero (*from scratch*) em um esquema de validação cruzada com 10 pastas (otimizador ADAM, lotes de tamanho 6 e taxa de aprendizado $\alpha = 0,005$); (iii) ajuste fino (*fine-tuning*) em cima dos pesos da ImageNet ($\alpha = 0,00005$). Nesse caso, as imagens foram dimensionadas baseado no tamanho da camada de entrada de cada modelo de CNN, isto é, 299x299x3 e 224x224x3 pixels para o InceptionV3 e ResNet50, respectivamente.

A Tabela 28 apresenta uma compilação dos resultados de classificação para todos os métodos e em ambos os formatos de imagem. Primeiramente, em relação aos métodos aplicados em imagens em níveis de cinza, a maior acurácia de 95,3% foi atingido pelo método PCLNN usando o esquema LOO e a imagem no tamanho original. Em seguida, os métodos CLBP (86,5%), ERCNN¹ (85,1%) e GLDM (85,1%) alcançaram os melhores desempenhos. Testes com as imagens reduzidas (224x224 pixels) também foram realizados, entretanto, para esse caso, os métodos propostos obtiveram um desempenho claramente inferior.

Em um segundo experimento, os métodos são avaliados com as imagens no formato RGB. Nesse caso, os métodos desenvolvidos para texturas em níveis de cinza foram adaptados para imagens em RGB usando a estratégia integrativa, isto é, os métodos são aplicados em cada canal de cor e depois os vetores dos canais são concatenados para formar a representação final. Quando comparado com as DCNNs, os métodos SSRNN e PLCNN alcançaram as melhores acurácias. O método proposto para imagem RGB, SSRNN, consegue uma acurácia alta mesmo em imagens reduzidas (224x224). Por outro lado, as CNNs obtêm os melhores resultados quando aplicadas como extratores nas imagens RGB com a resolução total, ao invés de treiná-las usando imagens redimensionadas. De fato, embora os modelos treinados do zero (*from scratch*) ou ajustados consigam 100% de acurácia no treinamento em muitos casos, a acurácia reportada para o teste é consideravelmente inferior. Isso evidencia a dificuldade de treinar modelos complexos em pequenos conjuntos de dados, o que leva a *overfitting* e considerável instabilidade (maior

desvio padrão entre iterações de validação cruzada).

Tabela 28 – Resultado de classificação para diferentes métodos na classificação de níveis de poluente em imagens de Jacaranda Caroba em formatos em escala de cinza e RGB, e com diferentes tamanhos. Os resultados foram obtidos com o classificador LDA, enquanto para os métodos CNNs que foram treinados, os resultados são dados pelo próprio classificador da rede neural.

Métodos	Tamanho da Imagem	Acurácia (%)	
		LOO	10-Pastas
Nível de Cinza			
GLDM	2048x2048x1	85,1	83,7 (±1, 2)
CLBP	2048x2048x1	86,5	87,2 (±2, 4)
Fourier	2048x2048x1	68,2	65,0 (±2, 7)
RGB			
Opponent-Gabor	2048x2048x3	81,1	84,7 (±2, 3)
SSN	2048x2048x3	82,4	83,0 (±2, 5)
SSN	224x224x3	74,3	74,9 (±2, 5)
ResNet50	224x224x3	71,6	72,2 (±2, 2)
ResNet50	2048x2048x3	89,9	89,4 (±1, 2)
InceptionV3	299x299x3	63,5	67,1 (±2, 4)
InceptionV3	2048x2048x3	87,8	86,9 (±1, 5)
CNNs - <i>From scratch</i>			
ResNet50	224x224x3	59,6 (±5, 3)	
InceptionV3	299x299x3	55,1 (±3, 5)	
CNNs - <i>Fine-tuning</i>			
ResNet50	224x224x3	83,1 (±2, 7)	
InceptionV3	299x299x3	81,8 (±3, 6)	
Métodos Propostos - Nível de cinza			
ERCNN ¹	224x224	73,0	69,1 (±1, 8)
ERCNN ²	224x224	58,8	60,4 (±2, 4)
PLCNN	224x224	61,5	62,4 (±1, 9)
ERCNN ¹	2048x2048x1	85,1	81,7 (±2, 5)
ERCNN ²	2048x2048x1	74,3	72,6 (±1, 7)
PLCNN	2048x2048x1	95,3	91,7 (±1, 4)
Métodos Propostos - RGB			
ERCNN ¹ (integrativo)	224x224x3	70,9	71,7 (±2, 7)
ERCNN ² (integrativo)	224x224x3	79,1	77,1 (±2, 5)
PLCNN (integrativo)	224x224x3	79,1	80,7 (±2, 0)
SSRNN ¹	224x224x3	91,2	91,7 (±1, 4)
SSRNN ²	224x224x3	93,9	91,5 (±1, 9)
ERCNN ¹ (integrativo)	2048x2048x3	77,0	77,0 (±1, 8)
ERCNN ² (integrativo)	2048x2048x3	86,5	84,9 (±2, 1)
PLCNN (integrativo)	2048x2048x3	91,2	91,2 (±1, 2)
SSRNN ¹	2048x2048x3	95,3	94,1 (±2, 3)
SSRNN ²	2048x2048x3	93,2	93,2 (±1, 4)

Fonte: Elaborada pelo autor.

Em relação aos métodos propostos, pode-se perceber que as imagens com resolução total fornecem as melhores acurácias, entretanto, os resultados com resolução reduzida são encorajadores, por exemplo, o SSRNN¹ e SSRNN², atingem acurácias de 91,2% e 93,9%, respectivamente. Esses resultados com resolução consideravelmente inferior são interessantes, pois isso reduz

drasticamente o custo computacional, permitindo que o método seja uma opção interessante para aplicações em tempo real. Tal desempenho também confirma que essa abordagem proposta especificamente para imagens coloridas é mais adequada do que uma abordagem integrativa, que não considera informações espectrais. Porém, é importante ressaltar os resultados do método proposto PLCNN, que em ambos os formatos de imagem obteve resultados comparáveis aos demais métodos originalmente propostos para RGB.

Além disso, pode-se perceber que o método proposto SSRNN melhora, significativamente, o desempenho comparado ao método SSN, que também utiliza a modelagem espaço-espectral. Isso pode ser explicado pela efetividade das representações aprendidas em caracterizar as propriedades espaciais e espectrais inerentes das redes complexas modeladas, em contraste com as medidas estatísticas utilizadas em trabalhos anteriores. A [Figura 54](#) apresenta a matriz de confusão para o resultado de 95,3% obtido pelo método SSRNN¹ nas imagens RGB. Como pode-se notar, a confusão na distinção acontece entre imagens expostas a concentrações de fluoreto de potássio que são próximas, por exemplo, duas amostras da concentração 15mg/L foram classificadas incorretamente como 5mg/L. Por outro lado, o método conseguiu diferenciar corretamente o caso controle (sem exposição nenhuma) do restante dos casos que foram expostos, indicando mudanças morfológicas na plasticidade foliar quando as folhas são expostas a concentrações de fluoreto de potássio. Por fim, ressalta-se que todos os métodos propostos utilizaram os parâmetros padrões definidos anteriormente, sem ajustes para as imagens estudadas. Portanto, é esperado que ajustes nos parâmetros para essas imagens também pode melhorar o desempenho.

8.3 Diagnóstico de Câncer de Próstata por meio de Imagens de Genossensores

Nos últimos anos, o desenvolvimento de técnicas de detecção tem atraído o interesse de pesquisadores de diversas áreas da ciência devido à necessidade de triagem e testes rápidos para um diagnóstico precoce de infecções e doenças, aumentando, assim, a possibilidade de cura dos pacientes. A pandemia de COVID-19 também demonstrou ser essencial métodos de detecção rápidos com baixo custo de manufatura, sem a necessidade de profissionais altamente treinados para operar equipamentos e também fundamentado por uma análise de dados robusta ([RODRIGUES et al., 2021](#)). Essas questões e a urgência por novas técnicas também se aplicam ao câncer de próstata, uma doença geralmente assintomática, em que a detecção tardia faz com que esta seja uma das principais causas de morte em homens em países industrializados ([IBAU; ARSHAD; GOPINATH, 2017](#); [ZADRA et al., 2010](#)). Para tanto, trabalhos têm investigado vários tipos de biossensores para a obtenção de testes de baixo custo e a exploração de novos nanomateriais. Basicamente, biossensores são dispositivos analíticos que possuem componentes biológicos (e.g., antígenos, anticorpos etc.) que interagem com um analito e produz uma resposta

Figura 54 – Matriz de confusão para a classificação de níveis de poluente em imagens de Jacaranda Caroba. Essa matriz é referente ao melhor resultado de 95,3% obtido com o método proposto SSRNN¹ usando LOO e com a imagem RGB.

	5mg/L	15mg/L	30mg/L	60mg/L	1162mg/L	Controle
5mg/L	25	0	0	0	0	0
15mg/L	2	24	0	0	0	0
30mg/L	0	1	23	0	0	0
60mg/L	0	0	0	23	1	0
1162mg/L	0	0	0	3	21	0
Controle	0	0	0	0	0	25

Fonte: Elaborada pelo autor.

capturada por um detector físico-químico. Um dos principais biossensores são os genossensores, que são produzidos com nanotecnologia para detectar o agente que causa a infecção ou o material genético do paciente (DNA ou RNA).

Nesta seção, uma investigação para o uso de métodos de texturas em níveis de cinza é apresentada para analisar imagens de genossensores visando a detecção de câncer de próstata. Neste estudo, publicado em [Rodrigues et al. \(2021\)](#) com a colaboração de pesquisadores de diversas instituições, foi apresentado genossensores que podem ser construídos com um processo de fabricação simples e também aplicado com vários princípios de detecção. Dentre esses princípios, os métodos de análise de imagens e aprendizado de máquina foram aplicados em imagens microscópicas dos genossensores, demonstrando ser uma promissora metodologia para detecção e diagnóstico de câncer de próstata. Assim, é possível obter um diagnóstico rápido obtendo uma imagem de uma unidade de detecção exposta à amostra e processando essa imagem. Ao contrário das abordagens padrão que envolvem análise de imagem da própria amostra biológica, este trabalho usa a imagem da unidade de detecção. Essa estratégia funciona porque é bem conhecido que as superfícies dos biossensores são alteradas durante as medições ([RODRIGUES et al., 2021](#)). Até onde se sabe, este trabalho ([RODRIGUES et al., 2021](#)) é o primeiro que aplica análise de texturas e aprendizado de máquina nas imagens de genossensores para fins de detecção. As próximas seções descrevem os dados, experimentos e resultados desta investigação.

8.3.1 Dados

As imagens utilizadas neste trabalho foram capturadas por Microscopia Eletrônica de Varredura (em inglês, *Scanning Electron Microscopy* (SEM)) utilizando um dispositivo DSM 960 da Zeiss, Alemanha. No total, 32 imagens foram coletadas, correspondentes a unidades de detecção que foram submetidas a distintas concentrações de PCA3 além da sequência negativa (não complementar) e uma medição em branco (solução de PBS sem biomarcador) para controle. (RODRIGUES *et al.*, 2021). Isso leva a dois problemas: (i) binário, em que as amostras submetidas a concentrações de PCA3 devem ser separadas do controle; e (ii) multi-classe, em que cada uma das concentrações representa uma classe, além dos controles. As imagens em escala de cinza foram capturadas em duas escalas diferentes (200 nm e 300 nm) e depois redimensionadas em 1024×768 pixels. Em um procedimento de aumento de dados, as imagens foram cortadas em janelas quadradas sem sobreposição de 300×300 pixels, resultando na base de dados descrita na Tabela 29. Exemplos de imagens de cada classe são apresentados na Figura 55.

Figura 55 – Exemplos de imagens de cada classe capturadas com SEM de unidades sensoriais.

Fonte: Rodrigues *et al.* (2021).

Tabela 29 – Descrição do número de amostras da base de dados com imagens de genossensores.

Tamanho (pixels)	Total	Amostras por classe							
		n	o	p_0p00001	p_0p0001	p_0p001	p_0p01	p_0p1	p_1p0
300	192	12	18	24	24	24	36	30	24

Fonte: Adaptada de Rodrigues *et al.* (2021).

8.3.2 Experimentos e Resultados

Neste estudo, os dois métodos propostos para análise de texturas em escala de cinza (PLCNN e ERCNN²) foram aplicados nas imagens capturadas por SEM. Como a base de dados possui classes desbalanceadas, um processo de reamostragem foi empregado usando 100 configurações aleatórias (ou seja, 100 tentativas) com o mesmo número de exemplos sorteados para classe em cada configuração. Na etapa de classificação, os métodos foram avaliados usando três classificadores (LDA, SVM e 1-NN) em um esquema de validação cruzada com 10 pastas

para separar os conjuntos de treino e teste. Como medida de desempenho, a acurácia média e o desvio padrão foram calculados das 100 configurações aleatórias.

Os resultados para cada combinação de técnica de análise de textura e classificador nos dois casos estudados são reportados na [Tabela 30](#). Como pode-se observar, para o problema binário, a maioria dos métodos de análise de textura atingem acurácias elevadas com os três classificadores. A acurácia máxima foi de 99,9% ($\pm 0,3$) usando o método proposto PLCNN com classificadores SVM e LDA, indicando forte capacidade em distinguir as imagens das unidades sensoriais expostas ao PCA3 (com todas as concentrações juntas) das que não foram expostas (negativas e em branco). Isso também reforça que existem mudanças nas características texturais e morfológicas nas imagens das unidades de detecção.

No procedimento de classificação multi-classe, é possível perceber que as acurácias diminuem. Isso pode ser explicado pelas mudanças serem pequenas na baixas concentrações de PCA3, como constatado em inspeção visual das imagens. A melhor combinação também foi baseada no método proposto LCFNN e no algoritmo LDA que atingiu uma acurácia de 88,3% ($\pm 3,4$). De modo geral, também pode ser notado que as combinações dos classificadores SVM e LDA com os métodos LCFNN, CNRNN e GLDM são as mais adequadas para separar os grupos.

Tabela 30 – Acurácia e desvio padrão (em parênteses) para os experimentos binário e multi-classe da base de dados composta por imagens de biossensores para diagnóstico de câncer de próstata.

Métodos	Binário			Multi-classe		
	LDA	SVM	1-NN	LDA	SVM	1-NN
GLDM	92,8 (6,7)	98,7 (1,3)	98,5 (1,3)	75,7 (4,4)	79,3 (3,3)	62,8 (3,8)
Fourier	92,4 (5,5)	97,1 (1,9)	98,6 (1,2)	56,3 (5,7)	75,9 (3,9)	67,2 (5,2)
CNTD	98,1 (1,8)	96,9 (1,9)	97,3 (2,0)	68,6 (5,3)	75,5 (3,4)	62,6 (4,1)
Fractal	92,9 (5,8)	97,9 (1,5)	96,3 (2,9)	59,5 (5,2)	64,2 (4,0)	52,7 (4,8)
AHP	98,9 (1,3)	98,8 (1,3)	98,4 (1,3)	66,7 (6,2)	78,2 (3,7)	68,4 (4,9)
LBP	99,0 (1,3)	98,6 (1,4)	94,6 (3,3)	76,5 (4,7)	74,7 (3,5)	72,5 (4,7)
Métodos Propostos						
ERCNN ²	98,1 (1,7)	98,9 (1,0)	98,9 (1,0)	72,7 (4,3)	82,4 (3,5)	70,4 (3,7)
PLCNN	99,9 (0,3)	99,9 (0,3)	99,5 (0,9)	88,3 (3,4)	86,9 (3,1)	80,3 (3,8)

Fonte: Adaptada de [Rodrigues et al. \(2021\)](#).

De modo a visualizar o comportamento dos grupos, uma análise não-supervisionada dos vetores de características também foi realizada usando o *t-Distributed Stochastic Neighbor Embedding* (t-SNE) ([MAATEN; HINTON, 2008](#)), que é uma técnica de visualização de dados de alta dimensão amplamente utilizada devido à sua capacidade de revelar a estrutura dos dados. Assim, as características extraídas pelos quatro melhores métodos de textura foram projetadas em um espaço bidimensional usando o t-SNE, como pode ser visto na [Figura 56](#), com as amostras (pontos) rotuladas de acordo com o problema binário apenas para fins de visualização. Note que as classes formam grupos distintos (em preto para negativo e vermelho para positivo), principalmente, para o descritor GLDM, relevando uma estrutura de cluster. Por outro lado, o

agrupamento das amostras positivas em vermelho é menos claro, como se deve esperar pelos resultados de classificação supervisionada, que mostraram ser mais difícil separar as diferentes concentrações de PCA3. Isso também é corroborado pelas acurácias obtidas com o algoritmo não-supervisionado de agrupamento k-means, como é apresentado na [Tabela 31](#). Nesse experimento, as amostras são agrupadas pelo k-means sem informações de classe, indicando a capacidade de discriminação dos métodos de textura. Portanto, pode-se concluir que os resultados da análise de imagens SEM das unidades de genossensores demonstram que as técnicas de análise de texturas propostas podem ser uma opção poderosa para complementar os princípios elétricos, eletroquímicos e ópticos de detecção.

Figura 56 – Projeção t-SNE em 2D das assinaturas calculadas pelos métodos CNRNN, AHP, GLDM e LCFNN. Cada ponto representa uma amostra e a cor indica a classe para fins de visualização.

Fonte: Adaptada de [Rodrigues et al. \(2021\)](#).

8.4 Detecção de Osteoartrite Precoce de Joelho por meio de Imagens de raio-X

A OsteoArtrite (OA) é uma doença degenerativa que causa dor, inflamação e disfunções nas articulações ([MURRAY et al., 2013](#); [LOESER et al., 2013](#)), e afeta principalmente as mãos, joelhos e quadris, tornando uma das causas para a limitação de mobilidade físicas de pessoas após os 50 anos ([BIJLSMA; BERENBAUM; LAFEVER, 2011](#); [LESPRESSAILLES;](#)

Tabela 31 – Acurácia para os grupos calculados com k-means nos casos binário e multi-classe na base de dados de genossensores.

Métodos	Acurácia (%)	
	Binário	Multi-classe
GLDM	86,98	52,10
Fourier	87,50	62,50
CNTD	84,37	52,60
Fractal	90,10	48,44
AHP	80,21	46,35
LBP	82,81	51,56
Métodos Propostos		
ERCNN	95,83	52,10
PLCNN ²	87,50	70,83

Fonte: Adaptada de [Rodrigues et al. \(2021\)](#).

[JENNANE, 2012](#)). No joelho, a OA é caracterizada por mudanças na estrutura óssea, degradação da cartilagem, redução da largura do espaço articular e também pela formação de osteófitos ([BIJLSMA; BERENBAUM; LAFEVER, 2011](#)). Os estágios da OA podem ser categorizados seguindo a classificação proposta por Kellgren e Lawrence (K&L), que é baseada em características radiológicas, tais como, estreitamento da cartilagem articular e formação de osteófitos ([KELLGREN; LAWRENCE, 1957](#)). Os cinco estágios de gravidade segundo essa classificação são: normal (grau 0), duvidosa (grau 1), mínima (grau 2), moderada (grau 3) e grave (grau 4). Assim, o grau 0 indica a ausência definitiva de OA, enquanto o grau 2 refere-se a OA precoce ([KELLGREN; LAWRENCE, 1957](#)). Diversos estudos têm mostrado que a análise da textura do osso trabecular em imagens de raio-X pode ajudar a prever precocemente OA do joelho, isto é, antes da redução da largura articular e formação de osteófito ([BUCKLAND-WRIGHT, 2004](#); [JANVIER et al., 2017](#)). Para a detecção precoce de OA de joelho, imagens de raios-X de joelho normal são consideradas como controle (K& L grau 0) e OA mínima (K& L grau 2).

Tendo em vista as considerações acima, esta seção apresenta a aplicação de métodos de análise texturas em imagens de raio-X para detecção precoce de OA em joelho. As imagens de raio-X são adquiridas de forma simples e com baixo custo, além de ser facilmente compreendidas, o que é importante para uma aplicação robusta e barata. Este trabalho é fruto da cooperação com o grupo de pesquisa da Universidade de Orleans (França) liderado pelo prof. Rachid Jennane, que já resultou na seguinte publicação ([RIBAS et al., 2022](#)). As seções seguintes apresentam os dados, experimentos e os resultados.

8.4.1 Dados

As imagens utilizados nesta aplicação são da base de dados *Osteoarthritis Initiative* (OAI) (disponível em ([OAI..., 2018-11-14](#))), cujo objetivo é pesquisar a saúde de pacientes suscetíveis a OA de joelho ([LESTER, 2012](#)). Neste estudo, apenas radiografias computadorizadas, que são

separadas em diferentes categorias seguindo a classificação K&L, são levadas em consideração. Uma vez que o objetivo é a detecção precoce de OA de joelho, são consideradas os graus K&L 0, como caso controle (CC) ou joelhos normais, e K&L 2, para pacientes com Osteoartrite. Assim, radiografias de grau K&L 1, 3 e 4 são descartadas por não se enquadrar com a tarefa de detecção precoce. Portanto, o presente estudo utilizou uma base de dados composta por 688 radiografias de joelho, dividida em 344 amostras de indivíduos de controle e 344 de pacientes com OA. Para selecionar a região de interesse do osso (em inglês, *Region of Interest* (ROI)), uma abordagem semiautomática foi adotada baseado em quatro marcadores anatômicos (marcados em vermelho na Figura 58), definidos manualmente por um operador treinado. Em seguida, a distância geodésica é utilizada para estimar a borda tibial passando pelos marcadores anatômicos (JANVIER *et al.*, 2017). Por fim, é definido um ROI de 128×128 pixels, localizado na região tibial sob o osso cortical, e com ajuste horizontal determinado como o centro entre as extremidades (RIBAS *et al.*, 2022).

Figura 57 – Osso segmentado em que os pontos vermelhos representam os marcadores anatômicos e a linha tracejada representa a borda tibial.

Fonte: Ribas *et al.* (2022).

8.4.2 Experimentos e Resultados

Para a realização dos experimentos, o classificador SVM com kernel linear e um esquema de validação cruzada *leave-one-out* foram empregados. Como medida de desempenho, utilizou-se a acurácia, além da sensibilidade e a especificidade para medir a capacidade do método de detectar corretamente os casos positivos e negativos, respectivamente. Os resultados para os métodos propostos e outros métodos da literatura são apresentados na Tabela 32. Em relação aos descritores clássicos de textura, os métodos propostos obtiveram os melhores desempenhos seguido do método GLCM. A maior acurácia foi atingida pelo método proposto ERCNN³

Figura 58 – Exemplos de imagens ROI para K&L grau 0 (CC) na primeira linha e K&L grau 2 (OP) na segunda linha.

Fonte: OAI... (2018-11-14), Lester (2012).

(77,47%), que também obteve o melhor desempenho em termos de especificidade, o que é significativo no contexto de predição de OA de joelho.

No que diz respeito às técnicas baseadas em aprendizado profundo, na comparação foram utilizados os seguintes modelos de CNN: AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), VGG-16 (SIMONYAN; ZISSERMAN, 2014), GoogleNet (SZEGEDY *et al.*, 2015), InceptionV3 (SZEGEDY *et al.*, 2016) e Resnet50 (HE *et al.*, 2016). Esses modelos foram aplicados em dois experimentos diferentes: (i) *fine-tuning*: ajustando os pesos treinados na base de dados ImageNet para a base de OA e usando um esquema de validação cruzada de 10 pastas com 1 pasta para teste, 1 para validação e 8 para treinamento (todas as pastas foram usadas para teste); (ii) extração de características: as arquiteturas pré-treinadas na base ImageNet foram usadas para extrair as características e o GAP sobre os mapas de características da última camada convolucional são usados como descritores para o classificador SVM.

Como pode ser observado na Tabela 32, em ambos os experimentos, os métodos desenvolvidos alcançaram acurácias superiores às dos modelos de CNNs. A sensibilidade usando a abordagem ERCNN² é menor que a obtida usando o modelo AlexNet, enquanto a especificidade do ERCNN³ é ligeiramente inferior que a atingida pelo modelo ResNet50. Observe que, quando os modelos de CNNs selecionados são ajustados (Experimento 1), melhores resultados são alcançados em comparação com a abordagem que combina GAP e SVM (Experimento 2). Tal comportamento é de certa forma esperado, pois, os modelos de aprendizado profundo foram pré-treinados em um domínio não relacionado às imagens de OA. Assim, apenas características mais gerais são extraídas das imagens, tais como aquelas relacionadas às propriedades de forma e textura (RIBAS *et al.*, 2022).

Por fim, ressalta-se também que ambos os casos, controle e OA, tiveram K&L=0 no início do estudo, o que sugere que as características extraídas pelos métodos são, provavelmente, correspondentes as alterações ósseas subcondrais precoces que ocorrem antes da detecção radiológica definitiva de OA em radiografias simples do compartimento central da tíbia (RIBAS *et al.*, 2022). Outro ponto importante é que as imagens de raio-X foram adquiridas usando

diferentes dispositivos, indicando que os métodos não são dependentes do equipamento ou centro clínico. Portanto, de modo geral, os métodos desenvolvidos demonstraram ser uma boa opção para aplicação na detecção precoce de OA de joelhos.

Tabela 32 – Acurácias de classificação usando SVM para os métodos aplicados na base de OA.

	Métodos	Sensibilidade (%)	Especificidade (%)	Acurácia (%)
Descritores Clássicos	GLCM	79,36	75,00	77,18
	Fractal Fourier	76,16	72,67	74,42
	LBP	70,06	72,09	71,08
	CLBP	68,31	67,15	67,73
CNNs: <i>Fine-Tuning</i>	AlexNet	79,94	70,64	75,29
	VGG-16	73,98	71,22	73,98
	GoogleNet	78,20	72,38	75,29
	InceptionV3	78,20	67,15	72,67
	ResNet50	74,71	77,32	76,02
CNNs: GAP+SVM	AlexNet	79,07	69,19	74,13
	VGG-16	77,33	68,31	72,82
	GoogleNet	65,41	53,20	59,30
	InceptionV3	77,62	67,73	72,67
	ResNet50	72,09	63,66	67,88
Métodos Propostos	ERCNN ¹	79,07	75,29	77,18
	ERCNN ²	79,36	75,58	77,47
	ERCNN ³	77,91	77,03	77,47
	PCLNN	77,91	75,29	76,60

Fonte: Elaborada pelo autor.

8.5 Reconhecimento de Espécies de Plantas baseado em Análise de Textura Foliar

A identificação de espécies de plantas é um problema desafiador devido à sua variedade, complexidade intrínseca da natureza e características taxonômicas, tornando a tarefa de reconhecimento manual por biólogos e botânicos um processo demorado e sujeito a erros humanos. Uma das formas de realizar a identificação é através da análise das propriedades morfológicas e anatômicas das folhas, que podem ser facilmente encontradas e coletadas na natureza ou em centros urbanos (GONCALVES, 2013; CASANOVA, 2013).

Nesta seção, os métodos de análise de texturas são aplicados para identificação de espécies de plantas. Essa tarefa é desafiadora devido à variação de padrões texturais e de cor dentro de uma mesma espécie ou até mesmo de uma mesma folha, além da similaridade de características entre diferentes espécies. Além da sua natureza complexa, essas variações podem ser resultado de diversos fatores tais como exposição a chuva, clima, sol etc. Isso faz com que esse seja um problema desafiador para a área de visão computacional. Portanto, neste estudo, os métodos são testados para caracterizar a textura foliar de imagens escaneadas de plantas, detalhadas na seção de dados. Os experimentos e resultados também são discutidos a seguir.

8.5.1 Dados

A base de imagens de folhas deste estudo é dividida em 20 espécies de plantas com 20 amostras cada, conforme em [Casanova, Junior e Bruno \(2009\)](#). As plantas foram escaneadas individualmente por meio da digitalização de sua superfície com um scanner de mesa (resolução de 1200 dpi), ajustando o seu eixo central na posição vertical. Tais imagens podem sofrer de imperfeições causadas por sujeira na folha ou no vidro do scanner, e também do processo de digitalização que pode causar sombras ou sobreposições ([CASANOVA; JUNIOR; BRUNO, 2009](#)). Como o objetivo é analisar a microtextura foliar, de cada imagem três janelas não sobrepostas de 128×128 pixels foram recortadas manualmente, resultando em uma base de dados de 1200 imagens. Esse procedimento aumenta o desafio de classificação, pois existe variabilidade da textura na superfície de uma mesma planta. A [Figura 59](#) apresenta algumas amostras de diferentes classes da base de dados.

8.5.2 Experimentos e Resultados

Neste trabalho, os métodos de análise de texturas foram aplicados nas imagens convertidas em níveis de cinza, exceto o método proposto para texturas coloridas SSRNN e o método SSN que usaram a imagem RGB. O classificador LDA foi utilizado para realizar a classificação dos descritores calculados pelos métodos. O esquema de validação utilizado foi o *leave-one-out* com a separação das 3 janelas de um mesmo espécime para teste e as janelas restantes para treinamento do classificador, repetindo esse processo para todas as folhas. Dessa forma, janelas de uma mesma folha não são usadas simultaneamente no treino e teste.

Para medir o desempenho dos métodos, duas métricas distintas foram adotadas. Na primeira, a acurácia é dada pela porcentagem de folhas classificadas corretamente baseado nas três janelas recortadas dela. Para tanto, para cada janela de cada espécime, o LDA calcula a probabilidade posteriori da janela ser de uma dada classe, gerando um vetor de probabilidades igual ao número de classes ou espécies (20) da base. Assim, considerando às três janelas J_i^1, J_i^2, J_i^3 recortadas de uma mesma folha i , é calculado a média dos três vetores de probabilidades de classes dados pelo classificador LDA. Em seguida, a classe com a maior probabilidade média é atribuída para a folha i que originou as janelas. Essa estratégia, proposta por [Casanova, Junior e Bruno \(2009\)](#), visa minimizar o erro na identificação da espécie de uma folha devido à alta variabilidade de padrões texturais e morfológicos na superfície de uma mesma folha. A segunda estratégia empregada neste estudo é baseada na classificação individual das janelas, sendo a acurácia a porcentagem de janelas classificadas corretamente. Note que nesse caso, janelas de uma mesma folha podem ser classificadas em espécies diferentes, tornando o problema mais difícil e testando a robustez dos métodos de caracterização.

A [Tabela 33](#) apresenta os resultados para os métodos de textura na classificação das folhas e janelas. Em relação à classificação das folhas, o método proposto SSRNN¹ com uma acurácia de 96,25% supera todos os métodos comparados, inclusive o método SSN que também

Figura 59 – Exemplos de 4 amostras (colunas) de 7 classes (linhas) da base de dados de folha.

Fonte: [Casanova, Junior e Bruno \(2009\)](#).

utiliza as imagens em RGB. Também é possível observar que informações de cor são importantes para esse problema, fazendo com que o desempenho de reconhecimento seja sempre superior quando considerado os três canais de cores. Os demais métodos propostos para níveis de cinza também melhoram a acurácia em comparação aos outros, com o método PCLNN alcançando a maior acurácia (92,00%). Por outro lado, a partir da tabela, é evidente que a classificação das janelas individualmente é mais difícil. Nessa tarefa, o desempenho de muitos métodos diminuiu significativamente (em média 10%) comparado com classificação das folhas como, por exemplo, de 82,50% para 72,00% com o LBP, de 88,50% para 76,58% usando o LCP e de 83,00% para 66,17% no método BSIF. Esse comportamento é esperado devido a alta variação de padrões de texturas em folha da mesma espécie.

Os métodos propostos conseguiram atingir mais uma vez as maiores acurácias de classi-

ificação em ambos os tipos de imagem. O método proposto SSRNN¹ para imagens RGB obteve taxa de acerto de 94,08%, enquanto nas imagens em escala de cinza o método proposto ERCNN³ alcançou uma taxa de 87,25%. Diferentemente de outros métodos, é importante ressaltar que os métodos propostos tiveram uma redução menor nas acurácias na comparação entre os dois experimentos, com destaque para o método SSRNN que teve um decréscimo de cerca de 2% apenas. O método SSRNN também melhora a acurácia quando comparado ao SSN (mesma modelagem de RC), evidenciando a importância do aprendizado de representações para caracterizar as redes complexas. Por fim, os resultados demonstram a capacidade de caracterização e robustez dos métodos propostos mesmo em um problema real e complexo, tornando-os ferramentas interessantes para o desenvolvimento de sistemas automatizados de baixo custo computacional para identificar espécies de plantas.

Tabela 33 – Acurácias para diferentes métodos de texturas aplicados no problema de classificação de espécies (folhas e janelas) de plantas usando o classificador LDA. Os métodos foram executados nas imagens convertidas para escala de cinza, com exceção dos métodos SSRNN¹, SSRNN² e SSN que usaram a imagem RGB.

Métodos	Folhas (%)	Janelas (%)
AHP	88,75	79,17
BSIF	83,00	66,17
FLBP	89,00	82,75
GLCM	77,50	68,42
GLDM	88,00	79,92
LCP	88,50	76,58
LFD	85,50	74,67
LPQ	83,75	73,00
SLBP	89,50	80,08
Fractal Fourier	66,75	58,17
Fourier	75,00	65,75
Fractal	80,50	70,75
Gabor	86,75	77,25
LBP	82,50	72,00
LBPV	58,50	48,08
CNTD	89,50	83,33
SSN	95,50	91,83
Métodos Propostos		
ERCNN ¹	89,25	82,33
ERCNN ²	91,75	86,25
ERCNN ³	91,25	87,25
PLCNN	92,00	86,08
SSRNN ¹	96,25	94,08
SSRNN ²	95,50	93,67

Fonte: Elaborada pelo autor.

8.6 Considerações Finais

Este capítulo avaliou os métodos desenvolvidos nesta tese no âmbito prático. Para tanto, os métodos foram aplicados em cinco problemas reais relevantes. Os métodos propostos para análise de formas DTN e CRSA obtiveram bom desempenho para identificar espécies de peixe de diferentes famílias baseado no contorno do otólito. Em texturas em níveis de cinza, os métodos propostos conseguiram resultados promissores nos quatro problemas em que foram testados, superando os métodos da literatura na maioria dos casos. Em relação às texturas coloridas, os resultados em dois problemas desafiadores com imagens de Jacarandá Caroba e folhas de plantas confirmaram a eficácia do método proposto SSRNN na caracterização de padrões de textura e cor. Em todos os casos, é evidente que a modelagem direcionada de redes complexas combinada com a proposta de caracterização por meio de aprendizado de representações com RNNs melhora o desempenho de reconhecimento. Além disso, deve-se notar que mesmo sem otimização de parâmetros (usou o padrão previamente definido), os métodos conseguiram manter um bom desempenho nos problemas estudados. Finalmente, embasado nos resultados reportados, foi possível concluir que os métodos desenvolvidos ao longo deste trabalho podem contribuir cientificamente com outras áreas da ciência, bem como na solução de problemas relevantes da sociedade.

CONCLUSÃO

A modelagem e caracterização de redes complexas permanecem como tópicos de pesquisa desafiadores, sendo amplamente investigados por cientistas, apesar dos diversos trabalhos que têm sido reportados ao longo das duas últimas décadas. Em relação à modelagem, o desafio é construir modelos que mapeiem as propriedades intrínsecas dos dados em redes complexas, sem utilizar muitos parâmetros e de forma rápida. Por outro lado, a respeito da caracterização, a pesquisa visa desenvolver medidas e métodos que consigam capturar as informações topológicas da rede complexa toda, de modo que a sua complexidade possa ser descrita por uma nova representação.

Nesse contexto, esta tese tem justamente como objetivo investigar e desenvolver técnicas de modelagem e caracterização de redes complexas focado na aplicação em problemas clássicos e importantes de visão computacional. Em termos de modelagem, abordagens inovadoras visando construir redes complexas direcionadas dinâmicas foram desenvolvidas para o problema de texturas em três tipos diferentes de dados: imagens em níveis de cinza, RGB e vídeos. Para as texturas em níveis de cinza, o modelo proposto mapeia os pixels em vértices e utiliza informações de intensidade de cinza e distância para definir o conjunto de arestas ponderadas com direção. Em seguida, em texturas coloridas e dinâmicas, essa abordagem é estendida com a incorporação de propriedades importantes para esses problemas, tais como, informações espectrais, espaciais e de movimento. Esses três modelos compartilham das mesmas propriedades: apenas um parâmetro para construir a RC (raio máximo da dinâmica) e arestas direcionadas baseado na direção do gradiente. Embora sejam simples e otimizadas, essas modelagens permitem gerar redes complexas que representam os principais padrões presentes nos diferentes tipos de texturas.

Por meio das redes complexas construídas pelos diversos modelos, novas formas de caracterização foram investigadas. A principal proposta nesse sentido foi a introdução de representações aprendidas pela arquitetura de redes neurais randomizadas (RNNs) para caracterizar as redes complexas. Isso foi feito usando conjuntos de informações topológicas das redes com-

plexas, como grau e força dos vértices, para treinar as RNNs, que possui um algoritmo de aprendizado simples e rápido, e produzir assim representações baseadas nos pesos treinados da camada de saída. Combinando essa ideia com as redes complexas geradas pelos modelos desenvolvidos, duas abordagens, ERCNN e PCLNN, foram propostas para analisar texturas em níveis de cinza. Em análise de texturas coloridas, também foi proposta uma representação espaço-espectral aprendida a partir de redes complexas direcionadas espaço-espectral. Os resultados experimentais e comparações com métodos da literatura em diversas bases de dados de texturas em cinza e coloridas, indicaram que as abordagens propostas produzem representações altamente discriminativas e competitivas em diferentes aspectos tal como acurácia de classificação, tempo de processamento, tamanho do vetor de atributos e robustez a transformações.

A partir desses estudos iniciais foi possível entender melhor as nuances existentes na combinação das metodologias de redes complexas e RNNs, inspirando assim o desenvolvimento de técnicas para os problemas de texturas dinâmicas e análise de contorno de formas. Para texturas dinâmicas, os estudos iniciais envolveram a aplicação das RNNs diretamente em três planos ortogonais dos vídeos para aprender representações. Em um segundo estudo, as RNNs foram empregadas para caracterizar as redes complexas que, por sua vez, modelaram as propriedades de aparência e movimento das texturas dinâmicas. Os resultados experimentais mostraram que em alguns casos a combinação de redes complexas e RNNs pode melhorar a descrição das texturas dinâmicas. Além disso, ambas as abordagens conseguiram alcançar um bom equilíbrio entre o tamanho do vetor de características, complexidade de tempo e acurácia de classificação, tornando-as comparáveis aos métodos mais discriminativos presentes na literatura.

No problema de análise de formas, as redes complexas foram combinadas com as RNNs e a EDT, dando origem a dois descritores do contorno de formas: CRSA e DTN. No descritor de redes de transformada da distância, embora adote medidas clássicas de redes complexas para caracterização, a capacidade de análise das formas é potencializada pela combinação da modelagem de redes complexas com o poder de análise de complexidade da transformada da distância. Nessa abordagem, os raios de dilatação resultantes da transformada da distância são modelados e analisados, ao invés da própria forma. Essa combinação realça as características intrínsecas da forma, incrementando, consideravelmente, o desempenho de classificação em muitos casos. Em seguida, uma segunda abordagem propõe uma nova estratégia para caracterizar a dinâmica de evolução das redes complexas que mapeiam as formas. Para isso, essa técnica emprega as RNNs para aprender representações a partir de mapas de características topológicas gerados da rede complexa modelada. Os resultados demonstraram que a abordagem proposta para caracterizar a dinâmica de evolução da rede complexa consegue melhorar significativamente o desempenho na classificação de formas em comparação as outras técnicas. Além disso, verificou-se que ambas as propostas conseguem garantir propriedades importantes e desejáveis em várias aplicações que envolvem a análise de formas, tais como, invariância à escala e rotação, e robustez a ruído.

Respalado pelo desempenho médio geral, pode-se afirmar que as RNNs melhoraram a caracterização das redes complexas em relação a outras medidas e técnicas clássicas utilizadas em trabalhos anteriores. Em alguns casos, como em análise de formas e texturas coloridas, os métodos desenvolvidos foram comparados com outros métodos que utilizam uma modelagem similar dos dados. Isto sugere que a rede complexa consegue mapear e incorporar na sua topologia a complexidade e aspectos físicos das imagens, entretanto, muitas vezes essas informações não são adequadamente extraídas ou caracterizadas pela medida empregada. Portanto, os resultados alcançados indicaram que a proposta de aprendizado de representações produz características mais discriminativas, demonstrando a importância no desenvolvimento de técnicas mais sofisticadas para caracterizar as redes complexas. Essa abordagem de caracterização pode inclusive ser estendida e explorada para reconhecimento de padrões em outros tipos de redes complexas que não modelam imagens.

Ainda no contexto de caracterização de redes complexas, um estudo foi realizado com a finalidade de melhorar a extração de características a partir de padrões de evolução temporal (TEPs) gerados pela técnica LLNA. Ao contrário das técnicas desenvolvidas que focam em análise de imagens, neste estudo, a caracterização das redes complexas foi explorada de forma mais ampla, visando classificar redes complexas gerais de larga escala. Portanto, a abordagem proposta pode ser vista como uma ferramenta mais universal de reconhecimento de padrões que pode ser empregada para caracterizar e classificar redes complexas de diversos problemas. O estudo realizado é uma extensão do LLNA, com a incorporação de padrões binários para descrever os padrões de evolução temporal e, consequentemente, as redes complexas. Tal incorporação melhora a descrição das redes complexas em comparação ao método anterior, como foi verificado nos resultados de classificação de diversos tipos de redes complexas. Isso faz com que a proposta seja uma opção interessante para estudar outros TEPs de autômato celulares.

Os métodos teóricos propostos ao longo deste trabalho possibilitaram o desenvolvimento de aplicações em problemas reais, cumprindo, assim, outro objetivo desta tese. Isso permitiu avaliar o desempenho em termos práticos e a aplicabilidade dos métodos em outras áreas da ciência, servindo como um termômetro da relevância da pesquisa. Para tanto, os métodos desenvolvidos foram aplicados em cinco problemas envolvendo as áreas de biologia, botânica, físico-química e médica. Embasado nos resultados reportados, é possível verificar que os métodos desenvolvidos são robustos e atingem um bom desempenho em relação aos problemas estudados, mesmo utilizando os parâmetros padrão definidos em bases de dados de referência, tornando-os opções interessantes para a aplicação rápida em diversas áreas.

À luz dos resultados alcançados, de forma mais ampla, acredita-se que este trabalho oferece uma contribuição relevante para a área de redes complexas e com o novo e promissor campo de pesquisa que estuda como conectar redes neurais artificiais e redes complexas para reconhecimento de padrões. Tais resultados motivam investigações futuras que explorem modelos de redes complexas e arquiteturas mais sofisticadas de RNN para reconhecimento

de padrões. Além disso, os métodos desenvolvidos com base nessas ferramentas proporciona importantes contribuições para as áreas de visão computacional estudadas (análise de texturas em níveis de cinza, coloridas e dinâmicas, e análise de formas), motivando pesquisas futuras com essas metodologias. Por fim, a multidisciplinaridade inerente do trabalho também resultou em contribuições em termos práticos para as áreas de visão computacional, biologia, botânica, físico-química e médica.

9.1 Trabalhos Futuros

Os resultados reportados neste trabalho demonstraram o enorme potencial das abordagens, motivando também investigações futuras de diversos aspectos das metodologias. Alguns trabalhos futuros sugeridos que podem ser relevantes são:

- utilizar arquiteturas de redes neurais randomizadas mais sofisticadas com mais de uma camada oculta e com mais de um neurônio de saída;
- estender a abordagem para aprendizado de representações com RNNs para redes complexas de outros tipos de dados;
- estender a abordagem de análise de texturas coloridas em imagens RGB para imagens hiperespectrais;
- empregar o método LLNA-BP para caracterização de redes complexas de imagens;
- otimizar a busca por regras de transição do LLNA-BP;
- aprofundar o estudo da aplicação de análise de imagens de biossensores.

9.2 Trabalhos Gerados

As contribuições diretas e adicionais desta tese resultaram em artigos científicos que estão publicados, submetidos ou em elaboração. Esses artigos são listados a seguir.

Publicados:

- RIBAS, L. C.; NEIVA, MARIANE B.; BRUNO, ODEMIR M.. Distance Transform Network for Shape Analysis. *Information Sciences*, v. 470, p. 28-42, 2018.
- RIBAS, L. C.; SÁ JUNIOR, J. J. M.; SCABINI, L. F.; BRUNO, O. M. Fusion of complex networks and randomized neural networks for texture analysis. *Pattern Recognition*, v. 103, p. 107189, 2020.

- RIBAS, L. C.; MACHICAO, J.; BRUNO, O. M. Life-like network automata descriptor based on binary patterns for network classification. *Information Sciences*, v. 515, p. 156–168, 2020.
- RIBAS, L. C.; SÁ JUNIOR, J. J. M.; MANZANERA, A.; BRUNO, O. M. Learning graph representation with Randomized Neural Network for dynamic texture classification. *Applied Soft Computing*, v. 114, p. 108035, 2022.
- RIBAS, L. C.; RIAD, RABIA; JENNANE, RACHID; BRUNO, ODEMIR M.. A complex network based approach for knee Osteoarthritis detection: Data from the Osteoarthritis initiative. *Biomedical Signal Processing and Control*, v. 71, p. 103133, 2022.
- RIBAS, L. C.; BRUNO, O. M.. Dynamic texture analysis using networks generated by deterministic partially self-avoiding walks. *Physica A: Statistical Mechanics and its Applications*, v. 541, p. 122105, 2020.
- RIBAS, L. C.; GONÇALVES, W. N.; BRUNO, O. M. Dynamic texture analysis with diffusion in networks. *Digital Signal Processing*, v. 92, p. 109–126, 2019.
- RIBAS, L. C.; ARRUDA, MAURO S.; SILVA, JONATHAN A.; CASTRO JR., AMAURY A.; BRUNO, ODEMIR M.; GONÇALVES, WESLEY N.. Fractal Dimension of Bag of Visual Words. *Pattern Analysis and Applications*, v. 21, p. 1, 2018.
- SCABINI, L. F.; RIBAS, L. C.; BRUNO, O. M. Spatio-spectral networks for color-texture analysis. *Information Sciences*, v. 515, p. 64 – 79, 2020.
- SÁ JR, JARBAS J. M.; RIBAS, L. C.; BRUNO, ODEMIR M.. Randomized neural network based signature for dynamic texture classification. *Expert Systems with Applications*, v. 135, p. 194–200, 2019.
- SCABINI, L. F.; RIBAS, L. C.; NEIVA, M. B.; JUNIOR, A. G.; FARFÁN, A. J.; BRUNO, O. M.. Social interaction layers in complex networks for the dynamical epidemic modeling of COVID-19 in Brazil. *Physica A: Statistical Mechanics and its Applications*, v. 564, p. 125498, 2021.
- MACHICAO, JEANETH ; RIBAS, L. C.; SCABINI, LEONARDO F.S.; BRUNO, ODEMIR M.. Cellular automata rule characterization and classification using texture descriptors. *Physica A: Statistical Mechanics and its Applications*, v. 497, p. 109–117, 2018.
- RODRIGUES, V. C.; SOARES, J. C.; SOARES, A. C.; BRAZ, D. C.; MELENDEZ, M. E.; RIBAS, L. C.; SCABINI, L. F.; BRUNO, O. M.; CARVALHO, A. L.; REIS, R. M. et al. Electrochemical and optical detection and machine learning applied to images of genosensors for diagnosis of prostate cancer with the biomarker PCA3. *Talanta*, v. 222, p. 121444, 2021.

- SOARES, J. C., SOARES, A. C., RODRIGUES, V. C., OITICICA, P. R. A., RAYMUNDO-PEREIRA, P. A., BOTT-NETO, J. L., BUSCAGLIA, L., CASTRO, L. D. C., RIBAS, L. C. et al. Detection of a SARS-CoV-2 sequence with genosensors using data analysis based on information visualization and machine learning techniques. *Materials Chemistry Frontiers*, v. 5(15), p. 5658-5670, 2021.
- RIBAS, L. C.; BRUNO O. M.. Dynamic Texture Classification Using Deterministic Partially Self-avoiding Walks on Networks. *International Conference on Image Analysis and Processing*, pp. 82-93, 2019. Springer, Cham.
- RIBAS, L. C.; MANZANERA, A.; BRUNO, O. M.. A Fractal-Based Approach to Network Characterization Applied to Texture Analysis. *International Conference on Computer Analysis of Images and Patterns*. pp. 129-140, 2019. Springer, Cham.
- RIBAS, L. C.; BRUNO, ODEMIR M.. Deterministic Partially Self-Avoiding Walks on Networks for Natural Shapes Classification. *23rd Iberoamerican Congress on Pattern Recognition - CIARP*, 2018, Madrid/Spain.
- SCABINI, L. F.; CONDORI, R. H.; RIBAS, L. C.; BRUNO, O. M.. Evaluating Deep Convolutional Neural Networks as Texture Feature Extractors. *International Conference on Image Analysis and Processing*, pp. 192-202, 2019, Springer, Cham.

Submetidos/Elaboração:

- RIBAS, L. C.; SÁ JUNIOR, J. J. M.; SCABINI, L. F.; BRUNO, O. M.. Learning Local Complex Features using Randomized Neural Networks for Texture Analysis. arXiv preprint arXiv:2007.05643
- RIBAS, L. C.; SCABINI, L. F.; CONDORI, R. H.; BRUNO, O. M.. Color-Texture Analysis using Spatio-Spectral Representation Learned with Randomized Neural Networks.
- RIBAS, L. C.; BRUNO, ODEMIR M.. Learning Complex Representations using Randomized Neural Networks on Topological Features Maps for Shape Classification.
- SCABINI, L. F.; RIBAS, L. C.; RIBEIRO, E.; BRUNO, O. M.. Deep Topological Embedding with CNNs for Complex Network Classification.
- SCABINI, L. F.; RIBAS, L. C., GROSSO, A.; FERNANDES, B. S. D.; BRUNO, O. M.. Network Science for Hyperspectral Image Analysis: Characterizing the Jacaranda Caroba Leaf Plasticity for Environmental Biosensing.
- CONDORI, R. H.; RIBAS, L. C.; Biagioni, S.; BEHLING, H.; BRUNO, O. M.. Fast detection and classification of pollen grains through convolutional neural networks.

REFERÊNCIAS

- ABDELMOUNAIME, S.; DONG-CHEN, H. New brodatz-based image databases for grayscale color and multiband texture analysis. **International Scholarly Research Notices**, Hindawi, v. 2013, 2013. Citado nas páginas 218 e 222.
- AKIMUSHKIN, C.; AMANCIO, D. R.; OLIVEIRA JR., O. N. Text authorship identified using the dynamics of word co-occurrence networks. **PLOS ONE**, Public Library of Science, v. 12, n. 1, p. 1–15, 01 2017. Citado nas páginas 30, 38 e 153.
- ALOM, M. Z.; TAHA, T. M.; YAKOPCIC, C.; WESTBERG, S.; HASAN, M.; ESESN, B. C. V.; AWWAL, A. A. S.; ASARI, V. K. The history began from alexnet: A comprehensive survey on deep learning approaches. **CoRR**, abs/1803.01164, 2018. Citado nas páginas 55, 56, 57 e 58.
- AMANCIO, D. R.; COMIN, C. H.; CASANOVA, D.; TRAVIESO, G.; BRUNO, O. M.; RODRIGUES, F. A.; COSTA, L. da F. A systematic comparison of supervised classifiers. **PloS one**, Public Library of Science, v. 9, n. 4, p. e94137, 2014. Citado na página 150.
- ANDREARCZYK, V.; WHELAN, P. F. Using filter banks in convolutional neural networks for texture classification. **Pattern Recognition Letters**, Elsevier, v. 84, p. 63–69, 2016. Citado na página 58.
- _____. Convolutional neural network on three orthogonal planes for dynamic texture classification. **Pattern Recognition**, Elsevier, v. 76, p. 36–49, 2018. Citado nas páginas 104, 117, 118 e 119.
- ANDRUSHIA, A. D.; THANGARAJAN, R. Rts-elm: an approach for saliency-directed image segmentation with ripplet transform. **Pattern Analysis and Applications**, Springer, v. 23, n. 1, p. 385–397, 2020. Citado na página 51.
- ARASHLOO, S. R.; KITTLER, J. Dynamic Texture Recognition Using Multiscale Binarized Statistical Image Features. **IEEE Transactions on Multimedia**, v. 16, n. 8, p. 2099–2109, 2014. Citado na página 118.
- ARII, K.; PARROTT, L. Examining the colonization process of exotic species varying in competitive abilities using a cellular automaton model. **Ecological Modelling**, Elsevier BV, v. 199, n. 3, p. 219–228, dec 2006. Citado na página 39.
- ARRIETA, A. B.; DÍAZ-RODRÍGUEZ, N.; SER, J. D.; BENNETOT, A.; TABIK, S.; BARBADO, A.; GARCÍA, S.; GIL-LÓPEZ, S.; MOLINA, D.; BENJAMINS, R. *et al.* Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. **Information Fusion**, Elsevier, v. 58, p. 82–115, 2020. Citado na página 140.
- BACCIU, D.; COLOMBO, M.; MORELLI, D.; PLANS, D. Randomized neural networks for preference learning with physiological data. **Neurocomputing**, v. 298, p. 9 – 20, 2018. Citado na página 51.
- BACKES, A. R. **Estudos de métodos de análise de complexidade em imagens**. Tese (Doutorado em Ciências Matemáticas e de Computação) — ICMC - Universidade de São Paulo, São Carlos, 2010. Citado nas páginas 30 e 225.

BACKES, A. R.; BRUNO, O. M. Shape classification using complex network and multi-scale fractal dimension. **Pattern Recognition Letters**, v. 31, n. 1, p. 44–51, 2010. Citado nas páginas [43](#) e [121](#).

_____. Shape skeleton classification using graph and multi-scale fractal dimension. In: SPRINGER. **International Conference on Image and Signal Processing**. [S.l.], 2010. p. 448–455. Citado na página [121](#).

_____. Texture analysis using volume-radius fractal dimension. **Applied Mathematics and Computation**, v. 219, n. 11, p. 5870–5875, 2013. Citado na página [38](#).

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. A complex network-based approach for boundary shape analysis. **Pattern Recognition**, v. 42, n. 1, p. 54–67, 2009. Citado nas páginas [45](#), [122](#), [125](#), [126](#), [131](#), [134](#), [138](#), [144](#), [147](#), [148](#), [222](#) e [227](#).

_____. Plant leaf identification based on volumetric fractal dimension. **International Journal of Pattern Recognition and Artificial Intelligence**, World Scientific, v. 23, n. 06, p. 1145–1160, 2009. Citado nas páginas [73](#), [87](#) e [175](#).

_____. Color texture analysis based on fractal descriptors. **Pattern Recognition**, v. 45, n. 5, p. 1984–1992, 2012. Citado nas páginas [77](#), [217](#) e [220](#).

_____. Texture analysis and classification: A complex network-based approach. **Information Sciences**, Elsevier, v. 219, p. 168–180, 2013. Citado nas páginas [43](#), [73](#), [88](#), [90](#), [101](#) e [217](#).

BACKES, A. R.; FLORINDO, J. B.; BRUNO, O. M. Shape analysis using fractal dimension: A curvature based approach. **Chaos**, v. 22, n. 4, p. 043103(1–8), 2012. Citado na página [121](#).

BACKES, A. R.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis based on maximum contrast walker. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, NY, USA, v. 31, n. 12, p. 1701–1707, set. 2010. ISSN 0167-8655. Citado nas páginas [45](#) e [122](#).

_____. Texture analysis using graphs generated by deterministic partially self-avoiding walks. **Pattern Recognition**, v. 44, n. 8, p. 1684–1689, 2011. ISSN 0031-3203. Citado na página [43](#).

BAI, X.; LATECKI, L. J. Path similarity skeleton graph matching. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 30, n. 7, p. 1282–1292, 2008. Citado na página [121](#).

BALDI, P.; POLLASTRI, G. The principled design of large-scale recursive neural network architectures-DAG-RNNs and the protein structure prediction problem. **Journal of Machine Learning Research**, v. 4, n. 4, p. 575–602, 2004. ISSN 1532-4435. Citado na página [59](#).

BANERJEE, A.; JOST, J. Spectral plot properties: Towards a qualitative classification of networks. **NHM**, v. 3, n. 2, p. 395–411, 2008. Citado nas páginas [30](#) e [153](#).

BARABÁSI, A.; PÓSFAL, M. **Network Science**. [S.l.]: Cambridge University Press, 2016. ISBN 9781107076266. Citado na página [35](#).

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999. Citado nas páginas [29](#), [35](#), [36](#) e [160](#).

BARABÁSI, A. L.; OLTVAI, Z. N. Network biology: understanding the cell's functional organization. **Nature reviews genetics**, Nature Publishing Group, v. 5, n. 2, p. 101–113, 2004. Citado nas páginas 29 e 35.

BARESI, L.; HECKEL, R. Tutorial introduction to graph transformation: A software engineering perspective. In: CORRADINI, A.; EHRIG, H.; KREOWSKI, H.-J.; ROZENBERG, G. (Ed.). **ICGT**. [S.l.]: Springer, 2002. (Lecture Notes in Computer Science, v. 2505), p. 402–429. Citado na página 59.

BIASOTTI, S.; MARINI, S.; SPAGNUOLO, M.; FALCIDIENO, B. Sub-part correspondence by structural descriptors of 3d shapes. **Computer-Aided Design**, Elsevier, v. 38, n. 9, p. 1002–1019, 2006. Citado na página 121.

BIJLSMA, J. W.; BERENBAUM, F.; LAFEVER, F. P. Osteoarthritis: an update with relevance for clinical practice. **The Lancet**, v. 377, n. 9783, p. 2115 – 2126, 2011. ISSN 0140-6736. Citado nas páginas 181 e 182.

BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks: Structure and dynamics. **Physics reports**, Elsevier, v. 424, n. 4, p. 175–308, 2006. Citado na página 37.

BORNHOLDT, S.; SCHUSTER, H. G. **Handbook of graphs and networks: from the genome to the internet**. [S.l.]: John Wiley & Sons, 2006. Citado na página 35.

BREMM, C.; SCHULZ, U. Otolith atlas of fish of the sinos river. **Brazilian Journal of Biology**, SciELO Brasil, v. 74, p. 274–282, 2014. Citado na página 169.

BRODATZ, P. **Textures: A photographic album for artists and designers**. [S.l.]: Dover Publications, 1966. Citado nas páginas 15, 77, 91, 217 e 218.

BRUNA, J.; ZAREMBA, W.; SZLAM, A.; LECUN, Y. Spectral Networks and Locally Connected Networks on Graphs. **CoRR**, abs/1312.6203, p. 1–14, 2013. Disponível em: <<http://arxiv.org/abs/1312.6203>>. Citado nas páginas 31, 58 e 59.

BUCKLAND-WRIGHT, C. Subchondral bone changes in hand and knee osteoarthritis detected by radiography. **Osteoarthritis and Cartilage**, v. 12, n. Supplement, p. 10 – 19, 2004. ISSN 1063-4584. The Role of Bone in the Treatment of Osteoarthritis: Proceedings of the 2001 Coral Gables Symposium. Citado na página 182.

CALVETTI, D.; MORIGI, S.; REICHEL, L.; SGALLARI, F. Tikhonov regularization and the L-curve for large discrete ill-posed problems. **Journal of Computational and Applied Mathematics**, v. 123, n. 1, p. 423 – 446, 2000. Citado na página 52.

CANTERO, S. V. A. B.; GONCALVES, D. N.; SCABINI, L. F. dos S.; GONCALVES, W. N. Importance of vertices in complex networks applied to texture analysis. **IEEE transactions on cybernetics**, IEEE, v. 50, n. 2, p. 777–786, 2018. Citado nas páginas 29, 43, 44 e 61.

CASANOVA, D. **Redes complexas em visão computacional com aplicações em bioinformática**. Tese (Doutorado em Física Aplicada) — Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2013. Citado nas páginas 29, 30, 35, 43, 46, 185 e 225.

CASANOVA, D.; JUNIOR, J. J. de M. S.; BRUNO, O. M. Plant leaf identification using gabor wavelets. **International Journal of Imaging Systems and Technology**, Wiley Subscription Services, Inc., A Wiley Company, v. 19, n. 3, p. 236–243, 2009. ISSN 1098-1098. Citado nas páginas 186 e 187.

CERNADAS, E.; FERNÁNDEZ-DELGADO, M.; GONZÁLEZ-RUFINO, E.; CARRIÓN, P. Influence of normalization and color space to color texture classification. **Pattern Recognition**, Elsevier, v. 61, p. 120–138, 2017. Citado na página 94.

CHALUMEAU, T.; COSTA, L. d.; LALIGANT, O.; MERIAUDEAU, F. Texture discrimination using hierarchical complex networks. In: DAMIANI, E.; YATONGNON, K.; SCHELKENS, P.; DIPANDA, A.; LEGRAND, L.; CHBEIR, R. (Ed.). **Signal Processing for Image Enhancement and Multimedia Processing**. [S.l.]: Springer US, 2008, (Multimedia Systems and Applications Series, v. 31). p. 95–102. Citado na página 43.

CHALUMEAU, T.; COSTA, L. d. F.; LALIGANT, O.; MERIAUDEAU, F. Texture discrimination using hierarchical complex networks. In: **International Conference on Signal Image Technology and Internet based Systems (SITIS)**. [S.l.]: Springer US, 2006. p. 543–550. Citado na página 43.

CHAN, A. B.; VASCONCELOS, N. Classifying video with kernel dynamic textures. In: **CVPR**. [S.l.: s.n.], 2007. p. 1–6. Citado na página 117.

CHOPARD, B.; LUTHI, P.; DROZ, M. Reaction-diffusion cellular automata model for the formation of leisegang patterns. **Physical Review Letters**, American Physical Society (APS), v. 72, n. 9, p. 1384–1387, feb 1994. Citado na página 39.

CIMPOI, M.; MAJI, S.; KOKKINOS, I.; MOHAMED, S.; VEDALDI, A. **Describing textures in the wild**. 2014. Disponível em: <http://openaccess.thecvf.com/content_cvpr_2014/papers/Cimpoi_Describing_Textures_in_2014_CVPR_paper.pdf>. Acesso em: 25 jan. 2017. Citado na página 102.

CIMPOI, M.; MAJI, S.; KOKKINOS, I.; VEDALDI, A. Deep filter banks for texture recognition, description, and segmentation. **International Journal of Computer Vision**, Springer, v. 118, n. 1, p. 65–94, 2016. Citado nas páginas 58, 89, 94 e 102.

CORNEA, N. D.; SILVER, D.; MIN, P. Curve-skeleton properties, applications, and algorithms. **IEEE Transactions on visualization and computer graphics**, IEEE Computer Society, v. 13, n. 3, p. 0530–548, 2007. Citado na página 121.

COSTA, L. d. F.; BOAS, P. R. V.; SILVA, F.; RODRIGUES, F. A pattern recognition approach to complex networks. **Journal of Statistical Mechanics: theory and experiment**, v. 2010, n. 11, p. P11015, 2010. Citado na página 162.

COSTA, L. d. F.; RODRIGUES, F.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in Physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citado nas páginas 29, 35, 36, 38, 125 e 162.

COSTA, L. d. F. D.; JR, R. M. C. **Shape analysis and classification: theory and practice**. [S.l.]: CRC Press, Inc., 2000. Citado na página 121.

COUTO, L. N.; RIBEIRO, T. P.; BACKES, A. R.; BARCELOS, C. A. Z. Texture characterization via improved deterministic walks on image-generated complex network. In: **Image Processing (ICIP), 2015 IEEE International Conference on**. [S.l.: s.n.], 2015. p. 4416–4420. Citado na página 44.

COVER, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. **IEEE transactions on electronic computers**, IEEE, n. 3, p. 326–334, 1965. Citado na página 51.

CZARAN, T.; BARTHA, S. Spatiotemporal dynamic models of plant populations and communities. **Trends in Ecology & Evolution**, Elsevier BV, v. 7, n. 2, p. 38–42, feb 1992. Citado na página 39.

DAUGMAN, J.; DOWNING, C. Gabor wavelets for statistical pattern recognition. In: MIT PRESS. **The handbook of brain theory and neural networks**. [S.l.], 1995. p. 414–419. Citado na página 87.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: IEEE. **Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on**. [S.l.], 2009. p. 248–255. Citado nas páginas 89 e 101.

DERPANIS, K. G.; WILDES, R. P. Dynamic texture recognition based on distributions of space-time oriented structure. In: IEEE. **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [S.l.], 2010. p. 191–198. Citado na página 104.

_____. Classification of traffic video based on a spatiotemporal orientation analysis. In: **IEEE Workshop on Applications of Computer Vision (WACV)**. [S.l.: s.n.], 2011. p. 606–613. Citado na página 104.

Dimitropoulos, K.; Barmpoutis, P.; Grammalidis, N. Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 25, n. 2, p. 339–351, 2015. Citado na página 104.

DORETTO, G.; CHIUSO, A.; WU, Y.; SOATTO, S. Dynamic textures. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 51, n. 2, p. 91–109, 2003. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1023/A%3A1021669406132>>. Citado na página 104.

DOROGOVTSSEV, S. N.; MENDES, J. F. Evolution of networks. **Advances in physics**, Taylor & Francis, v. 51, n. 4, p. 1079–1187, 2002. Citado na página 160.

DUDEK, G. Generating random weights and biases in feedforward neural networks with random hidden nodes. **Information Sciences**, v. 481, p. 33 – 56, 2019. Citado na página 51.

ERDÖS, P.; RÉNYI, A. On random graphs. **Publicationes Mathematicae Debrecen**, v. 6, p. 290–297, 1959. Citado na página 159.

ERDÖS, P.; RÉNYI, A. On the evolution of random graphs. **Publ. Math. Inst. Hungar. Acad. Sci**, Citeseer, v. 5, p. 17–61, 1960. Citado nas páginas 29, 35 e 36.

EULER, L. Solutio problematis ad geometriam situs pertinentis. **Commentarii academiae scientiarum Petropolitanae**, v. 8, p. 128–140, 1741. Citado na página 35.

- EVERINGHAM, M.; ESLAMI, S. M. A.; GOOL, L. V.; WILLIAMS, C. K. I.; WINN, J.; ZISSERMAN, A. The pascal visual object classes challenge: A retrospective. **International Journal of Computer Vision**, v. 111, n. 1, p. 98–136, jan. 2015. Citado na página 54.
- EVERITT, B.; DUNN, G. **Applied Multivariate Data Analysis**. [S.l.]: Wiley, 2001. (A Hodder Arnold Publication). ISBN 9780340741221. Citado nas páginas 77, 144 e 225.
- FABBRI, R.; COSTA, L. D. F.; TORELLI, J. C.; BRUNO, O. M. 2d euclidean distance transform algorithms: A comparative survey. **ACM Computing Surveys (CSUR)**, ACM, v. 40, n. 1, p. 2, 2008. Citado na página 123.
- FALVO, M. **Método de mapeamento espaço-espectral em imagens multi-espectrais e sua aplicação em tecidos vegetais**. 155 p. Tese (Doutorado) — Instituto de Física de São carlos, Universidade de São Paulo, São Carlos, 2016. Citado nas páginas 173 e 174.
- FLORINDO, J. B.; BRUNO, O. M. Fractal descriptors based on Fourier spectrum applied to texture analysis. **Physica A: statistical Mechanics and its Applications**, Elsevier, v. 391, n. 20, p. 4909–4922, 2012. Citado nas páginas 73 e 87.
- FUKUSHIMA, K.; MIYAKE, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: **Competition and cooperation in neural nets**. [S.l.]: Springer, 1982. p. 267–285. Citado na página 53.
- GAMA, J.; FACELI, K.; LORENA, A.; CARVALHO, A. D. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: Grupo Gen - LTC, 2011. ISBN 9788521618805. Citado nas páginas 30, 49 e 50.
- GHANEM, B.; AHUJA, N. Maximum margin distance learning for dynamic texture recognition. In: **Proceedings of the 11th European Conference on Computer Vision: Part II**. Berlin, Heidelberg: Springer-Verlag, 2010. (ECCV'10), p. 223–236. ISBN 3-642-15551-0, 978-3-642-15551-2. Citado nas páginas 109, 115, 219 e 223.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2014. Citado na página 54.
- GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002. Citado na página 35.
- GONCALVES, W. N. **Análise de texturas estáticas e dinâmicas e suas aplicações em biologia e nanotecnologia**. Tese (Doutorado em Física Aplicada) — Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2013. Citado nas páginas 185, 226 e 227.
- GONÇALVES, W. N.; BACKES, A. R.; MARTINEZ, A. S.; BRUNO, O. M. Texture descriptor based on partially self-avoiding deterministic walker on networks. **Expert Systems with Applications**, v. 39, n. 15, p. 11818–11829, 2012. ISSN 0957-4174. Citado nas páginas 43 e 44.
- GONÇALVES, W. N.; BRUNO, O. M. Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. **Expert Systems with Applications**, v. 40, n. 11, p. 4283 – 4300, 2013. Citado nas páginas 106, 117, 118 e 119.

GONÇALVES, W. N.; MACHADO, B. B.; BRUNO, O. M. A complex network approach for dynamic texture recognition. **Neurocomputing**, v. 153, p. 211 – 220, 2015. Citado nas páginas 20, 45, 61, 68, 104, 105, 111, 116, 117, 118 e 119.

GONÇALVES, W. N.; SILVA, N. R. da; COSTA, L. da F.; BRUNO, O. M. Texture recognition based on diffusion in networks. **Information Sciences**, v. 364–365, p. 51–71, 2016. ISSN 0020-0255. Citado nas páginas 30, 35, 38 e 44.

GUANGLI, W.; LIPING, L.; CHEN, Z.; DENGTAI, T. Video abnormal event detection based on elm. In: IEEE. **2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)**. [S.l.], 2019. p. 367–371. Citado na página 51.

GUO, Y.; ZHAO, G.; PIETIKÄINEN, M. Texture classification using a linear configuration model based descriptor. In: CITESEER. **BMVC**. [S.l.], 2011. p. 1–10. Citado na página 88.

GUO, Z.; ZHANG, L.; ZHANG, D. A completed modeling of local binary pattern operator for texture classification. **IEEE Transactions on Image Processing**, IEEE, v. 19, n. 6, p. 1657–1663, 2010. Citado nas páginas 87 e 101.

_____. Rotation invariant texture classification using lbp variance (lbpv) with global matching. **Pattern recognition**, Elsevier, v. 43, n. 3, p. 706–719, 2010. Citado na página 87.

HARALICK, R. M. Statistical and structural approaches to texture. **Proceedings of the IEEE**, IEEE, v. 67, n. 5, p. 786–804, 1979. Citado nas páginas 73 e 87.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. [S.l.]: Artmed, 2007. ISBN 9788577800865. Citado na página 49.

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. Citado nas páginas 54, 55, 56, 89, 101, 175 e 184.

HEARST, M. A.; DUMAIS, S. T.; OSUNA, E.; PLATT, J.; SCHOLKOPF, B. Support vector machines. **IEEE Intelligent Systems and their Applications**, IEEE, v. 13, n. 4, p. 18–28, 1998. Citado nas páginas 144, 162 e 226.

HENAFF, M.; BRUNA, J.; LECUN, Y. Deep Convolutional Networks on Graph-Structured Data. **CoRR**, abs/1506.05163, p. 1–10, 2015. Citado na página 59.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. **Neural Computation**, v. 18, n. 7, p. 1527–1554, 2006. ISSN 0899-7667. Citado na página 54.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 53.

HOLMES, G.; DONKIN, A.; WITTEN, I. H. Weka: A machine learning workbench. In: IEEE. **Proceedings of ANZIS'94-Australian New Zealand Intelligent Information Systems Conference**. [S.l.], 1994. p. 357–361. Citado na página 114.

HOO-CHANG, S.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; YAO, J.; MOLLURA, D.; SUMMERS, R. M. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. **IEEE transactions on medical imaging**, NIH Public Access, v. 35, n. 5, p. 1285, 2016. Citado na página 58.

- HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982. Citado na página 50.
- HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Citado na página 56.
- HU, M. K. Visual pattern recognition by moment invariants. **IRE transactions on information theory**, IEEE, v. 8, n. 2, p. 179–187, 1962. Citado na página 121.
- HUANG, G.; LIU, Z.; WEINBERGER, K. Q. Densely connected convolutional networks. **CoRR**, abs/1608.06993, 2016. Disponível em: <<http://arxiv.org/abs/1608.06993>>. Citado nas páginas 54 e 56.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: theory and applications. **Neurocomputing**, Elsevier, v. 70, n. 1, p. 489–501, 2006. Citado nas páginas 31 e 51.
- IBAU, C.; ARSHAD, M. M.; GOPINATH, S. C. Current advances and future visions on bioelectronic immunosensing for prostate-specific antigen. **Biosensors and Bioelectronics**, Elsevier, v. 98, p. 267–284, 2017. Citado na página 177.
- JAIN, A.; HEALEY, G. A multiscale representation including opponent color features for texture recognition. **IEEE Transactions on Image Processing**, v. 7, n. 1, p. 124–128, 1998. Citado na página 101.
- JANVIER, T.; JENNANE, R.; VALERY, A.; HARRAR, K.; DELPLANQUE, M.; LELONG, C.; LOEUILLE, D.; TOUMI, H.; LESPESSAILLES, E. Subchondral tibial bone texture analysis predicts knee osteoarthritis progression: data from the osteoarthritis initiative: Tibial bone texture & knee OA progression. **Osteoarthritis and Cartilage**, v. 25, n. 2, p. 259–266, 2017. ISSN 1063-4584. Citado nas páginas 182 e 183.
- JI, H.; YANG, X.; LING, H.; XU, Y. Wavelet domain multifractal analysis for static and dynamic texture classification. **IEEE Transactions on Image Processing**, IEEE, v. 22, n. 1, p. 286–299, 2013. Citado na página 118.
- JUNIOR, J. J. d. M. S.; BACKES, A. R. Shape classification using line segment statistics. **Information Sciences**, Elsevier, v. 305, p. 349–356, 2015. Citado na página 145.
- JUNIOR, J. J. d. M. S.; BACKES, A. R.; BRUNO, O. M. Randomized neural network based descriptors for shape classification. **Neurocomputing**, Elsevier, v. 312, p. 201–209, 2018. Citado na página 51.
- _____. Randomized neural network based signature for color texture classification. **Multidimensional Systems and Signal Processing**, Springer, v. 30, n. 3, p. 1171–1186, 2019. Citado na página 101.
- JUNIOR, J. J. de M. S.; BACKES, A. R.; BRUNO, O. M. Randomized neural network based signature for classification of titanium alloy microstructures. In: MENDOZA, M.; VELASTÍN, S. (Ed.). **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications**. Cham: Springer International Publishing, 2018. p. 669–676. ISBN 978-3-319-75193-1. Citado na página 51.

JUNIOR, J. J. de M. S.; BACKES, A. R.; CORTEZ, P. C. Texture analysis and classification using shortest paths in graphs. **Pattern Recognition Letters**, v. 34, n. 11, p. 1314–1319, 2013. ISSN 0167-8655. Citado na página [44](#).

JUNIOR, J. J. M. S.; BACKES, A. R. ELM based signature for texture classification. **Pattern Recognition**, v. 51, p. 395–401, 2016. Citado nas páginas [19](#), [31](#), [73](#), [75](#), [88](#), [92](#), [93](#), [105](#), [107](#), [108](#) e [135](#).

KANEHISA, M.; SATO, Y.; KAWASHIMA, M.; FURUMICHI, M.; TANABE, M. KEGG as a reference resource for gene and protein annotation. **Nucleic Acids Research**, Oxford University Press (OUP), v. 44, n. D1, p. D457–D462, 2015. Citado na página [160](#).

KANNALA, J.; RAHTU, E. Bsif: Binarized statistical image features. In: IEEE. **Pattern Recognition (ICPR), 2012 21st International Conference on**. [S.l.], 2012. p. 1363–1366. Citado na página [88](#).

KAYA, L. K. Y.; TEKIN, R. A computer vision system for the automatic identification of butterfly species via gabor-filter-based texture features and extreme learning machine: Gf+ elm. **Tem Journal**, Citeseer, p. 13, 2013. Citado na página [51](#).

KELLGREN, J.; LAWRENCE, J. Radiological assessment of osteo-arthritis. **Annals of the rheumatic diseases**, BMJ Group, v. 16, n. 4, p. 494–502, 1957. Citado na página [182](#).

KRAHMER, E.; ERK, S. V.; VERLEG, A. Graph-Based Generation of Referring Expressions. **Computational Linguistics**, v. 29, n. 1, p. 53–72, 2003. Citado na página [59](#).

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. **Advances In Neural Information Processing Systems**, p. 1–9, 2012. ISSN 10495258. Citado nas páginas [54](#), [55](#), [57](#), [101](#) e [184](#).

LABELLE, L.; LAUZON, D.; KONRAD, J.; DUBOIS, E. Arithmetic coding of a lossless contour based representation of label images. In: IEEE. **Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on**. [S.l.], 1998. v. 1, p. 261–265. Citado na página [43](#).

LAM, W.-K.; LI, C.-K. Rotated texture classification by improved iterative morphological decomposition. **IEE Proceedings-Vision, Image and Signal Processing**, IET, v. 144, n. 3, p. 171–179, 1997. Citado na página [73](#).

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015. Citado nas páginas [31](#) e [59](#).

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural computation**, MIT Press, v. 1, n. 4, p. 541–551, 1989. Citado na página [53](#).

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado na página [53](#).

LEHMER, D. H. Mathematical methods in large scale computing units. **Annals Comp. Laboratory Harvard University**, v. 26, p. 141–146, 1951. Citado na página [135](#).

- LEIBE, B.; SCHIELE, B. Analyzing appearance and contour based methods for object categorization. In: IEEE. **Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on**. [S.l.], 2003. v. 2, p. II–409. Citado nas páginas 144, 221 e 226.
- LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. <<http://snap.stanford.edu/data>>. Citado na página 160.
- LESPESSAILLES, E.; JENNANE, R. Assessment of bone mineral density and radiographic parameter texture at the tibial subchondral bone. **Osteoporosis International**, v. 23, p. 871–6, 11 2012. Citado nas páginas 181 e 182.
- LESTER, G. The osteoarthritis initiative: A NIH public–private partnership. **HSS Journal**, v. 8, n. 1, p. 62–63, Feb 2012. ISSN 1556-3324. Citado nas páginas 182 e 184.
- Li, W.; Mahadevan, V.; Vasconcelos, N. Anomaly detection and localization in crowded scenes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 36, n. 1, p. 18–32, 2014. Citado na página 104.
- LIAO, S. X.; PAWLAK, M. On image analysis by moments. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 18, n. 3, p. 254–266, 1996. Citado na página 121.
- LIN, M.; CHEN, Q.; YAN, S. Network in network. **arXiv preprint arXiv:1312.4400**, arXiv, 2013. Citado nas páginas 58, 89, 102 e 175.
- LIU, C.; ZOPH, B.; NEUMANN, M.; SHLENS, J.; HUA, W.; LI, L.-J.; FEI-FEI, L.; YUILLE, A.; HUANG, J.; MURPHY, K. Progressive neural architecture search. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 19–34. Citado na página 56.
- LOESER, R. F.; GOLDRING, S. R.; SCANZELLO, C. R.; GOLDRING, M. B. Osteoarthritis: A disease of the joint as an organ. **Arthritis & Rheumatism**, v. 64, n. 6, p. 1697–1707, 2013. Citado na página 181.
- LONCARIC, S. A survey of shape analysis techniques. **Pattern recognition**, Elsevier, v. 31, n. 8, p. 983–1001, 1998. Citado na página 121.
- MAANI, R.; KALRA, S.; YANG, Y.-H. Noise robust rotation invariant features for texture classification. **Pattern Recognition**, Elsevier, v. 46, n. 8, p. 2103–2116, 2013. Citado na página 88.
- MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. **Journal of machine learning research**, v. 9, n. 11, 2008. Citado na página 180.
- MACHADO, B. B.; SCABINI, L. F.; ORUE, J. P. M.; ARRUDA, M. S. de; GONCALVES, D. N.; GONCALVES, W. N.; MOREIRA, R.; RODRIGUES-JR, J. F. A complex network approach for nanoparticle agglomeration analysis in nanoscale images. **Journal of Nanoparticle Research**, Springer, v. 19, n. 2, p. 65, 2017. Citado na página 30.
- MACHICAO, J.; BRUNO, O. M. A cryptographic hash function based on chaotic network automata. **Journal of Physics: Conference Series**, IOP Publishing, v. 936, p. 012058, dec 2017. Citado na página 39.

- MACHICAO, J.; CORRÊA, E. A.; MIRANDA, G. H. B.; AMANCIO, D. R.; BRUNO, O. M. Authorship attribution based on life-like network automata. **PLOS ONE**, Public Library of Science (PLoS), v. 13, n. 3, p. e0193703, mar 2018. Citado nas páginas 30, 38 e 153.
- MACHICAO, J.; FILHO, H. A.; LAHR, D. J. G.; BUCKERIDGE, M.; BRUNO, O. M. Topological assessment of metabolic networks reveals evolutionary information. **Scientific Reports**, Springer Nature America, Inc, v. 8, n. 1, 2018. Citado na página 153.
- MACHICAO, J.; MARCO, A. G.; BRUNO, O. M. Chaotic encryption method based on life-like cellular automata. **Expert Systems with Applications**, Elsevier BV, v. 39, n. 16, p. 12626–12635, nov 2012. Citado na página 39.
- MACHICAO, J.; RIBAS, L. C.; SCABINI, L. F.; BRUNO, O. M. Cellular automata rule characterization and classification using texture descriptors. **Physica A: Statistical Mechanics and its Applications**, Elsevier BV, v. 497, p. 109–117, may 2018. Citado na página 39.
- MÄENPÄÄ, T.; PIETIKÄINEN, M. Classification with color and texture: jointly or separately? **Pattern recognition**, Elsevier, v. 37, n. 8, p. 1629–1640, 2004. Citado na página 94.
- MALAR, E.; KANDASWAMY, A.; CHAKRAVARTHY, D.; DHARAN, A. G. A novel approach for detection and classification of mammographic microcalcifications using wavelet analysis and extreme learning machine. **Computers in biology and medicine**, Elsevier, v. 42, n. 9, p. 898–905, 2012. Citado na página 51.
- MANJUNATH, B. S.; MA, W.-Y. Texture features for browsing and retrieval of image data. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 18, n. 8, p. 837–842, 1996. Citado nas páginas 73 e 87.
- MARR, C.; HÜTT, M. Outer-totalistic cellular automata on graphs. **Physics Letters A**, v. 373, n. 5, p. 546–549, 2009. Citado na página 40.
- MASI, I.; WU, Y.; HASSNER, T.; NATARAJAN, P. Deep face recognition: A survey. In: IEEE. **2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)**. [S.l.], 2018. p. 471–478. Citado na página 54.
- MATAS, J.; MARIK, R.; KITTLER, J. On representation and matching of multi-coloured objects. In: IEEE. **Computer Vision, 1995. Proceedings., Fifth International Conference on**. [S.l.], 1995. p. 726–732. Citado na página 43.
- MAURER, C. R.; QI, R.; RAGHAVAN, V. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 25, n. 2, p. 265–270, 2003. Citado nas páginas 123 e 128.
- MCAULEY, J. J.; LESKOVEC, J. Learning to discover social circles in ego networks. In: **NIPS**. [S.l.: s.n.], 2012. v. 2012, p. 548–56. Citado na página 160.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, Dec 1943. Citado na página 49.
- MEHRI, A.; DAROONEH, A. H.; SHARIATI, A. The complex networks approach for authorship attribution of books. **Physica A: Statistical Mechanics and its Applications**, v. 391, n. 7, p. 2429 – 2437, 2012. Citado nas páginas 30 e 153.

- MINSKY, M.; PAPERT, S. A. **Perceptrons: An introduction to computational geometry**. [S.l.]: MIT press, 2017. Citado na página 49.
- MIRANDA, G. H.; MACHICAO, J.; BRUNO, O. M. An optimized shape descriptor based on structural properties of networks. **Digital Signal Processing**, Elsevier BV, v. 82, p. 216–229, 2018. Citado na página 30.
- MIRANDA, G. H. B.; MACHICAO, J.; BRUNO, O. M. Exploring spatio-temporal dynamics of cellular automata for pattern recognition in networks. **Scientific Reports**, v. 6, n. 37329, 2016. Citado nas páginas 29, 30, 39, 40, 41, 42, 153, 154, 160, 161, 162 e 164.
- MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. **Frontiers in plant science**, Frontiers, v. 7, p. 1419, 2016. Citado na página 58.
- MOKHTARIAN, F.; BOBER, M. **Curvature scale space representation: theory, applications, and MPEG-7 standardization**. [S.l.]: Springer Science & Business Media, 2013. v. 25. Citado na página 122.
- MORRIS, O.; LEE, M. J.; CONSTANTINIDES, A. Graph theory for image analysis: an approach based on the shortest spanning tree. **Communications, Radar and Signal Processing, IEEE Proceedings F**, v. 133, n. 2, p. 146–152, 1986. ISSN 0143-7070. Citado na página 43.
- MUDULI, D.; DASH, R.; MAJHI, B. Automated breast cancer detection in digital mammograms: A moth flame optimization based elm approach. **Biomedical Signal Processing and Control**, Elsevier, v. 59, p. 101912, 2020. Citado na página 51.
- MURRAY, C. J.; ABRAHAM, J.; ALI, M. K.; ALVARADO, M.; ATKINSON, C.; BADDOUR, L. M.; BARTELS, D. H.; BENJAMIN, E. J.; BHALLA, K.; BIRBECK, G. *et al.* The state of us health, 1990-2010: Burden of diseases, injuries, and risk factors. **JAMA**, v. 310, n. 6, p. 591–606, 2013. Citado na página 181.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th International Conference on Machine Learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814. Citado na página 55.
- NANNI, L.; GHIDONI, S.; BRAHNAM, S. Handcrafted vs. non-handcrafted features for computer vision classification. **Pattern Recognition**, Elsevier, v. 71, p. 158–172, 2017. Citado nas páginas 31 e 71.
- NASIRUZZAMAN, A.; POTA, H.; MAHMUD, M. Application of centrality measures of complex network framework in power grid. In: **IEEE. IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society**. [S.l.], 2011. p. 4660–4665. Citado nas páginas 29 e 35.
- NEWMAN, M. E.; PARK, J. Why social networks are different from other types of networks. **Physical Review E**, APS, v. 68, n. 3, p. 036122, 2003. Citado nas páginas 29 e 35.
- NIEPERT, M.; AHMED, M.; KUTZKOV, K. Learning Convolutional Neural Networks for Graphs. **CoRR**, abs/1605.05273, 2016. Citado nas páginas 31 e 59.
- OAI Online. 2018–11–14. <<https://data-archive.nimh.nih.gov/oai/>>. Citado nas páginas 182 e 184.

- OJALA, T.; MAENPAA, T.; PIETIKAINEN, M.; VIERTOLA, J.; KYLLONEN, J.; HUOVINEN, S. Outex-new framework for empirical evaluation of texture analysis algorithms. In: IEEE. **Object recognition supported by user interaction for service robots**. [S.l.], 2002. v. 1, p. 701–706. Citado nas páginas 77, 217 e 219.
- OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 24, n. 7, p. 971–987, 2002. Citado nas páginas 73, 87 e 175.
- OJANSIVU, V.; HEIKKILÄ, J. Blur insensitive texture classification using local phase quantization. In: SPRINGER. **International conference on image and signal processing**. [S.l.], 2008. p. 236–243. Citado nas páginas 87, 101 e 175.
- OSOWSKI, S. *et al.* Fourier and wavelet descriptors for shape recognition using neural networks—a comparative study. **Pattern Recognition**, Elsevier, v. 35, n. 9, p. 1949–1957, 2002. Citado na página 144.
- PANJWANI, D. K.; HEALEY, G. Markov random field models for unsupervised segmentation of textured color images. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 17, n. 10, p. 939–954, 1995. Citado na página 73.
- PAO, Y.-H.; PARK, G.-H.; SOBAJIC, D. J. Learning and generalization characteristics of the random vector functional-link net. **Neurocomputing**, Elsevier, v. 6, n. 2, p. 163–180, 1994. Citado nas páginas 31 e 50.
- PAO, Y.-H.; TAKEFUJI, Y. Functional-link net computing: theory, system architecture, and functionalities. **Computer**, IEEE, v. 25, n. 5, p. 76–79, 1992. Citado nas páginas 31 e 50.
- PARK, S. K.; MILLER, K. W. Random number generators: good ones are hard to find. **Communications of the ACM**, ACM, v. 31, n. 10, p. 1192–1201, 1988. Citado nas páginas 75 e 135.
- PEROZZI, B.; AL-RFOU, R.; SKIENA, S. Deepwalk: Online learning of social representations. In: **Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2014. (KDD '14), p. 701–710. Citado na página 59.
- PÉTERI, R.; FAZEKAS, S.; HUISKES, M. J. DynTex: A comprehensive database of dynamic textures. **Pattern Recognition Letters**, Elsevier Science Inc., New York, NY, USA, v. 31, n. 12, p. 1627–1632, set. 2010. Citado nas páginas 219 e 223.
- PICARD, R.; GRACZYK, C.; MANN, S.; WACHMAN, J.; PICARD, L.; CAMPBELL, L. **Vision texture database**. Media Laboratory, MIT, Cambridge, Massachusetts: [s.n.], 1995. Citado nas páginas 77, 218 e 221.
- PLOTZE, R. d. O.; FALVO, M.; PÁDUA, J. G.; BERNACCI, L. C.; VIEIRA, M. L. C.; OLIVEIRA, G. C. X.; BRUNO, O. M. Leaf shape analysis using the multiscale minkowski fractal dimension, a new morphometric method: a study with passiflora (passifloraceae). **Canadian Journal of Botany**, NRC Research Press, v. 83, n. 3, p. 287–301, 2005. Citado na página 122.
- POLANA, R.; NELSON, R. Temporal texture and activity recognition. In: SHAH, M.; JAIN, R. (Ed.). **Motion-Based Recognition**. Springer Netherlands, 1997, (Computational Imaging and Vision, v. 9). p. 87–124. ISBN 978-90-481-4870-7. Disponível em: <http://dx.doi.org/10.1007/978-94-015-8935-2_5>. Citado na página 104.

POLESZCZUK, J.; ENDERLING, H. A high-performance cellular automaton model of tumor growth with dynamically growing domains. **Applied Mathematics**, Scientific Research Publishing, Inc., v. 05, n. 01, p. 144–152, 2014. Citado na página 39.

PRATAMA, M.; ANGELOV, P. P.; LU, J.; LUGHOFFER, E.; SEERA, M.; LIM, C. P. A randomized neural network for data streams. In: **2017 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2017. p. 3423–3430. Citado nas páginas 51 e 119.

PURWINS, H.; LI, B.; VIRTANEN, T.; SCHLÜTER, J.; CHANG, S.-Y.; SAINATH, T. Deep learning for audio signal processing. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 13, n. 2, p. 206–219, 2019. Citado na página 31.

QI, X.; LI, C.-G.; ZHAO, G.; HONG, X.; PIETIKÄINEN, M. Dynamic texture and scene classification by transferring deep image features. **Neurocomputing**, Elsevier, v. 171, p. 1230–1241, 2016. Citado na página 104.

RAVICHANDRAN, A.; CHAUDHRY, R.; VIDAL, R. View-invariant dynamic texture recognition using a bag of dynamical systems. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2009. p. 1651–1657. ISSN 1063-6919. Citado nas páginas 104, 115 e 219.

RAZAVIAN, A. S.; AZIZPOUR, H.; SULLIVAN, J.; CARLSSON, S. Cnn features off-the-shelf: an astounding baseline for recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2014. p. 806–813. Citado na página 58.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 91–99. Citado na página 58.

RIBAS, L. C. **Análise de texturas dinâmicas baseada em sistemas complexos**. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, doi:10.11606/D.55.2017.tde-28072017-141204, 2017. Citado na página 61.

RIBAS, L. C.; BRUNO, O. M. Dynamic texture analysis using networks generated by deterministic partially self-avoiding walks. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 541, p. 122105, 2020. Citado nas páginas 46, 105, 111, 117, 118 e 119.

RIBAS, L. C.; GONÇALVES, W. N.; BRUNO, O. M. Dynamic texture analysis with diffusion in networks. **Digital Signal Processing**, Elsevier, v. 92, p. 109–126, 2019. Citado nas páginas 38, 39, 45, 68, 69, 104, 105, 111, 117, 118 e 119.

RIBAS, L. C.; MACHICAO, J.; BRUNO, O. M. Life-like network automata descriptor based on binary patterns for network classification. **Information Sciences**, Elsevier, v. 515, p. 156–168, 2020. Citado nas páginas 42, 154, 155, 158, 164, 165, 166 e 167.

RIBAS, L. C.; NEIVA, M. B.; BRUNO, O. M. Distance transform network for shape analysis. **Information Sciences**, v. 470, p. 28 – 42, 2019. Citado nas páginas 30, 122, 124, 127, 128, 129 e 130.

RIBAS, L. C.; RIAD, R.; JENNANE, R.; BRUNO, O. M. A complex network based approach for knee osteoarthritis detection: Data from the osteoarthritis initiative. **Biomedical Signal Processing and Control**, Elsevier, v. 71, p. 103133, 2022. Citado nas páginas 182, 183 e 184.

- RIBAS, L. C.; Sá Junior, J. J. M.; SCABINI, L. F.; BRUNO, O. M. Fusion of complex networks and randomized neural networks for texture analysis. **Pattern Recognition**, v. 103, p. 107189, 2020. Citado nas páginas [30](#), [51](#), [62](#), [64](#), [74](#), [76](#), [77](#), [78](#), [79](#), [80](#), [86](#), [87](#) e [91](#).
- RIBAS, L. C.; Sá Junior, J. J. de M.; MANZANERA, A.; BRUNO, O. M. Learning graph representation with randomized neural network for dynamic texture classification. **Applied Soft Computing**, v. 114, p. 108035, 2022. ISSN 1568-4946. Citado nas páginas [69](#), [105](#), [111](#), [112](#), [113](#), [114](#) e [115](#).
- RIBEIRO, T. P.; COUTO, L. N.; BACKES, A. R.; BARCELOS, C. A. Z. Texture characterization via automatic threshold selection on image-generated complex network. In: PARDO, A.; KITTLER, J. (Ed.). **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications**. Cham: Springer International Publishing, 2015. p. 468–476. ISBN 978-3-319-25751-8. Citado na página [46](#).
- RIVERA, A. R.; CHAE, O. Spatiotemporal directional number transitional graph for dynamic texture recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 37, n. 10, p. 2146–2152, Oct 2015. ISSN 0162-8828. Citado nas páginas [104](#) e [118](#).
- RODRIGUES, V. C.; SOARES, J. C.; SOARES, A. C.; BRAZ, D. C.; MELENDEZ, M. E.; RIBAS, L. C.; SCABINI, L. F.; BRUNO, O. M.; CARVALHO, A. L.; REIS, R. M. *et al.* Electrochemical and optical detection and machine learning applied to images of genosensors for diagnosis of prostate cancer with the biomarker pca3. **Talanta**, Elsevier, v. 222, p. 121444, 2021. Citado nas páginas [177](#), [178](#), [179](#), [180](#), [181](#) e [182](#).
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, p. 65–386, 1958. Citado nas páginas [13](#), [49](#) e [50](#).
- ROSENFELD, A.; PFALTZ, J. Sequential operations in digital picture processing. **Journal of the ACM**, v. 13, n. 4, p. 471–494, 1966. Citado na página [122](#).
- ROSIN, P. Training cellular automata for image processing. **IEEE Transactions on Image Processing**, Institute of Electrical and Electronics Engineers (IEEE), v. 15, n. 7, p. 2076–2087, jul 2006. Citado na página [39](#).
- RUMELHART, D. E.; HINTON, G. E.; MCCLELLAND, J. L. *et al.* A general framework for parallel distributed processing. **Parallel distributed processing: Explorations in the microstructure of cognition**, v. 1, n. 45-76, p. 26, 1986. Citado na página [53](#).
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986. Citado na página [50](#).
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015. Citado na página [54](#).
- Sá Junior, J. J. M.; RIBAS, L. C.; BRUNO, O. M. Randomized neural network based signature for dynamic texture classification. **Expert Systems with Applications**, Elsevier, v. 135, p. 194–200, 2019. Citado nas páginas [51](#), [105](#), [106](#), [108](#) e [110](#).

SAHA, B.; GUPTA, S.; PHUNG, D.; VENKATESH, S. Multiple task transfer learning with small sample sizes. **Knowledge and information systems**, Springer, v. 46, n. 2, p. 315–342, 2016. Citado na página 58.

SAISAN, P.; DORETTO, G.; WU, Y. N.; SOATTO, S. Dynamic texture recognition. In: **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001**. [S.l.: s.n.], 2001. v. 2, p. II–58–II–63 vol.2. Citado nas páginas 109, 219 e 224.

SALIMI, N.; LOH, K. H.; DHILLON, S. K.; CHONG, V. C. Fully-automated identification of fish species based on otolith contour: using short-time fourier transform and discriminant analysis (stft-da). **PeerJ**, PeerJ Inc., v. 4, p. e1664, 2016. Citado nas páginas 169, 170 e 171.

SCABINI, L. F.; CONDORI, R. H.; GONÇALVES, W. N.; BRUNO, O. M. Multilayer complex network descriptors for color–texture characterization. **Information Sciences**, Elsevier, v. 491, p. 30–47, 2019. Citado nas páginas 44, 46, 61, 65, 66, 94 e 101.

SCABINI, L. F.; CONDORI, R. H.; RIBAS, L. C.; BRUNO, O. M. Evaluating deep convolutional neural networks as texture feature extractors. In: SPRINGER. **International Conference on Image Analysis and Processing**. [S.l.], 2019. p. 192–202. Citado na página 54.

SCABINI, L. F.; FISTAROL, D. O.; CANTERO, S. V.; GONÇALVES, W. N.; MACHADO, B. B.; JR, J. F. R. Angular descriptors of complex networks: A novel approach for boundary shape analysis. **Expert Systems with Applications**, Elsevier, v. 89, p. 362–373, 2017. Citado nas páginas 30, 39, 43, 45 e 145.

SCABINI, L. F.; GONÇALVES, W. N.; JR, A. A. C. Texture analysis by bag-of-visual-words of complex networks. In: SPRINGER INTERNATIONAL PUBLISHING. **Iberoamerican Congress on Pattern Recognition**. [S.l.], 2015. p. 485–492. Citado nas páginas 46 e 65.

SCABINI, L. F.; RIBAS, L. C.; BRUNO, O. M. Spatio-spectral networks for color-texture analysis. **Information Sciences**, v. 515, p. 64 – 79, 2020. ISSN 0020-0255. Citado nas páginas 44, 45, 64, 65, 66, 67, 94, 101 e 102.

SCARSELLI, F.; GORI, M.; TSOI, A. C.; HAGENBUCHNER, M.; MONFARDINI, G. The graph neural network model. **IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council**, v. 20, n. 1, p. 61–80, 2009. ISSN 1941-0093. Citado na página 59.

SCHMIDT, W. F.; KRAAIJVELD, M. A.; DUIN, R. P. W. Feedforward neural networks with random weights. In: **Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems**. [S.l.: s.n.], 1992. p. 1–4. Citado nas páginas 31 e 50.

SEBASTIAN, T. B.; KLEIN, P. N.; KIMIA, B. B. Recognition of shapes by editing their shock graphs. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 26, n. 5, p. 550–571, 2004. Citado nas páginas 144, 220 e 225.

SHARMA, N.; GUPTA, A. K. Impact of time delay on the dynamics of seir epidemic model using cellular automata. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 471, p. 114–125, 2017. Citado na página 39.

SILVA, I. N. d.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. [S.l.]: Artliber Editora, 2010. Citado na página 49.

SILVA, N. R. da; BAETENS, J.; OLIVEIRA, M.; BAETS, B. d.; BRUNO, O. Classification of cellular automata through texture analysis. **Information Sciences**, v. 370–371, p. 33–49, 2016. Citado na página 39.

SILVA, N. R. da; WEEEN, P. V. der; BAETS, B. D.; BRUNO, O. M. Improved texture image classification through the use of a corrosion-inspired cellular automaton. **NEUROCOMPUTING**, v. 149, p. 1560–1572, 2015. Citado na página 39.

SIMON, P.; UMA, V. Deep learning based feature extraction for texture classification. **Procedia Computer Science**, Elsevier, v. 171, p. 1680–1687, 2020. Citado na página 31.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, arXiv, 2014. Citado nas páginas 56, 89, 101 e 184.

SÖDERKVIST, O. **Computer vision classification of leaves from swedish trees**. 2001. Citado nas páginas 144 e 222.

SRIVASTAVA, N.; HINTON, G. E.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 55.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Thirty-First AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2017. Citado nas páginas 31, 56 e 89.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015. Citado nas páginas 53, 54, 56, 101 e 184.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. Citado nas páginas 54, 56, 89, 94, 101, 175 e 184.

TIKHONOV, A. N. On the solution of ill-posed problems and the method of regularization. **Dokl. Akad. Nauk USSR**, v. 151, n. 3, p. 501–504, 1963. Citado na página 52.

TIWARI, D.; TYAGI, V. A novel scheme based on local binary pattern for dynamic texture recognition. **Computer Vision and Image Understanding**, v. 150, p. 58–65, 2016. Citado nas páginas 20, 104, 115, 117 e 118.

_____. Dynamic texture recognition based on completed volume local binary pattern. **Multidimensional Systems and Signal Processing**, v. 27, n. 2, p. 563–575, 2016. Citado nas páginas 106, 117 e 118.

TIWARI, R. D.; TYAGI, V. Dynamic texture recognition using multiresolution edge-weighted local structure pattern. **Computers & Electrical Engineering**, v. 62, p. 485–498, 2017. Citado nas páginas 20, 104, 117 e 118.

- TORRES, R. d. S.; FALCAO, A. X.; COSTA, L. d. F. A graph-based approach for multiscale shape analysis. **Pattern Recognition**, Elsevier, v. 37, n. 6, p. 1163–1174, 2004. Citado nas páginas 122 e 144.
- TRIPATHI, S.; ACHARYA, S.; SHARMA, R. D.; MITTAL, S.; BHATTACHARYA, S. Using deep and convolutional neural networks for accurate emotion classification on deap dataset. In: **AAAI**. [S.l.: s.n.], 2017. p. 4746–4752. Citado na página 55.
- VES, E. de; ACEVEDO, D.; RUEDIN, A.; BENAVENT, X. A statistical model for magnitudes and angles of wavelet frame coefficients and its application to texture retrieval. **Pattern Recognition**, v. 47, n. 9, p. 2925 – 2939, 2014. Citado na página 73.
- WALLACE, T. P.; WINTZ, P. A. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. **Computer Graphics and Image Processing**, Elsevier, v. 13, n. 2, p. 99–126, 1980. Citado na página 122.
- WANG, X.; FENG, B.; BAI, X.; LIU, W.; LATECKI, L. J. Bag of contour fragments for robust shape classification. **Pattern Recognition**, Elsevier, v. 47, n. 6, p. 2116–2125, 2014. Citado na página 121.
- WANG, Y.; HU, S. Exploiting high level feature for dynamic textures recognition. **Neurocomputing**, Elsevier, v. 154, p. 217–224, 2015. Citado na página 118.
- _____. Chaotic features for dynamic textures recognition. **Soft Computing**, Springer, v. 20, n. 5, p. 1977–1989, 2016. Citado na página 118.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. **nature**, Nature Publishing Group, v. 393, n. 6684, p. 440–442, 1998. Citado nas páginas 29, 35, 36, 126 e 160.
- WAXMAN, B. M. Routing of multipoint connections. **IEEE Journal on Selected Areas in Communications**, v. 6, n. 9, p. 1617–1622, 1988. Citado na página 160.
- WESZKA, J. S.; DYER, C. R.; ROSENFELD, A. A comparative study of texture measures for terrain classification. **IEEE transactions on Systems, Man, and Cybernetics**, IEEE, n. 4, p. 269–285, 1976. Citado nas páginas 87 e 175.
- WILDES, R. P.; BERGEN, J. R. Qualitative spatiotemporal analysis using an oriented energy representation. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2000. p. 768–784. Citado na página 104.
- WOLFRAM, S. Statistical mechanics of cellular automata. **Reviews of Modern Physics**, v. 55, n. 3, p. 601–644, 1983. Citado na página 39.
- _____. Cellular automata as models of complexity. **Nature**, v. 311, n. 5985, p. 419–424, 1984. Citado na página 39.
- WU, A.-C.; XU, X.-J.; WANG, Y.-H. Excitable greenberg-hastings cellular automaton model on scale-free networks. **Physical Review E**, v. 75, n. 3, p. 032901, 2007. Citado na página 40.
- WU, H.; GU, X. Max-pooling dropout for regularization of convolutional neural networks. **CoRR**, abs/1512.01400, 2015. Disponível em: <<http://arxiv.org/abs/1512.01400>>. Citado na página 55.

- WU, W.-Y.; WANG, M.-J. J. Detecting the dominant points by the curvature-based polygonal approximation. **CVGIP: Graphical Models and Image Processing**, Elsevier, v. 55, n. 2, p. 79–88, 1993. Citado na página 144.
- XIN, R.; ZHANG, J.; SHAO, Y. Complex network classification with convolutional neural network. **CoRR**, abs/1802.00539, 2018. Disponível em: <<http://arxiv.org/abs/1802.00539>>. Citado nas páginas 30, 31, 60 e 153.
- XU, Y.; HUANG, S.; JI, H.; FERMÜLLER, C. Scale-space texture description on SIFT-like textons. **Computer Vision and Image Understanding**, Elsevier, v. 116, n. 9, p. 999–1013, 2012. Citado nas páginas 117 e 118.
- XU, Y.; QUAN, Y.; LING, H.; JI, H. Dynamic texture classification using dynamic fractal analysis. In: **2011 International Conference on Computer Vision**. [S.l.: s.n.], 2011. p. 1219–1226. ISSN 1550-5499. Citado nas páginas 104 e 117.
- YOUNG, T.; HAZARIKA, D.; PORIA, S.; CAMBRIA, E. Recent trends in deep learning based natural language processing. **ieee Computational intelligence magazine**, IEEE, v. 13, n. 3, p. 55–75, 2018. Citado na página 31.
- ZADRA, G.; PRIOLO, C.; PATNAIK, A.; LODA, M. New strategies in prostate cancer: targeting lipogenic pathways and the energy sensor ampk. **Clinical Cancer Research**, AACR, v. 16, n. 13, p. 3322–3328, 2010. Citado na página 177.
- ZHANG, D.; YIN, J.; ZHU, X.; ZHANG, C. Network representation learning: A survey. **IEEE transactions on Big Data**, IEEE, v. 6, n. 1, p. 3–28, 2018. Citado na página 31.
- ZHANG, Y.; WU, J.; CAI, Z.; DU, B.; YU, P. S. An unsupervised parameter learning model for RVFL neural network. **Neural Networks**, 2019. Citado na página 51.
- ZHANG, Z.; LIU, S.; MEI, X.; XIAO, B.; ZHENG, L. Learning completed discriminative local features for texture classification. **Pattern Recognition**, v. 67, p. 263 – 275, 2017. ISSN 0031-3203. Citado na página 73.
- ZHAO, G.; PIETIKÄINEN, M. Dynamic texture recognition using volume local binary patterns. In: **Dynamical Vision**. [S.l.]: Springer Berlin Heidelberg, 2006. p. 165–177. Citado nas páginas 104, 117, 118 e 119.
- ZHAO, G.; PIETIKAINEN, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 29, n. 6, p. 915–928, 2007. Citado nas páginas 106, 117, 118 e 119.
- ZHAO, J.; YU, H.; LUO, J.; CAO, Z. W.; LI, Y. Complex networks theory for analyzing metabolic networks. **Chinese Science Bulletin**, v. 51, n. 13, p. 1529–1537, 2006. Citado na página 160.
- Zhao, X.; Lin, Y.; Heikkilä, J. Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection. **IEEE Transactions on Multimedia**, v. 20, n. 3, p. 552–566, 2018. Citado na página 104.
- ZHENJIANG, M. Zernike moment-based image shape analysis and its application. **Pattern Recognition Letters**, v. 21, n. 2, p. 169 – 177, 2000. ISSN 0167-8655. Citado nas páginas 121 e 144.

ZHOU, H.; LIPOWSKY, R. Dynamic pattern evolution on scale-free networks. **Proceedings of the National Academy of Sciences of the United States of America**, v. 102, n. 29, p. 10052–10057, 2005. Citado na página [40](#).

ZHOU, J.; CUI, G.; HU, S.; ZHANG, Z.; YANG, C.; LIU, Z.; WANG, L.; LI, C.; SUN, M. Graph neural networks: A review of methods and applications. **AI Open**, Elsevier, v. 1, p. 57–81, 2020. Citado nas páginas [59](#) e [60](#).

ZHU, Z.; YOU, X.; CHEN, C. P.; TAO, D.; OU, W.; JIANG, X.; ZOU, J. An adaptive hybrid pattern for noise-robust texture analysis. **Pattern Recognition**, Elsevier, v. 48, n. 8, p. 2592–2608, 2015. Citado nas páginas [88](#) e [175](#).

ZINOVIEVA, O.; ZINOVIEV, A.; PLOSHIKHIN, V.; ROMANOVA, V.; BALOKHONOV, R. A solution to the problem of the mesh anisotropy in cellular automata simulations of grain growth. **Computational Materials Science**, Elsevier BV, v. 108, p. 168–176, oct 2015. Citado na página [39](#).

RECONHECIMENTO DE PADRÕES E BASES DE DADOS

Este apêndice descreve em detalhes as bases de dados e técnicas de reconhecimento de padrões utilizadas neste trabalho.

A.1 Bases de dados

A.1.1 *Texturas*

Nesta seção, as bases de dados de texturas utilizadas neste trabalho para avaliar os métodos são apresentadas. Todas as bases de dados são de texturas coloridas em imagens RGB, com exceção da Brodatz que é em nível de cinza. Para os métodos para análise de textura em nível de cinza, as imagens RGB foram convertidas em escala de cinza eliminando informações de tonalidade e saturação, e mantendo a luminância.

- *Brodatz* ([BRODATZ, 1966](#)): essa base de dados é composta de texturas originárias do álbum Brodatz. Neste trabalho, assim como em ([BACKES; CASANOVA; BRUNO, 2013](#)), 1776 imagens de textura de 128×128 pixels divididas em 111 classes foram usadas. Exemplos de texturas de algumas classes são mostradas na [Figura 60](#).
- *Outex* ([OJALA et al., 2002](#)): essa base de dados foi proposta para avaliação empírica de métodos de caracterização de texturas, consistindo de imagens de materiais em diferentes iluminações. São 68 imagens coloridas de 746×538 da TC_Outex_00013 recortadas em 20 sub-imagens de tamanho de 128×128 pixels sem sobreposição. Assim, no total são 1360 amostras. Algumas amostras de diferentes classes são apresentadas na [Figura 61](#)
- *USPTex* ([BACKES; CASANOVA; BRUNO, 2012](#)): a base contém 2292 amostras de texturas naturais coloridas divididas em 191 classes, com 12 imagens por classes, e cada imagem

possui dimensão de 128×128 pixels. A [Figura 62](#) mostra exemplos de 30 classes da base de dados.

- *Vistex*: a base *Vision Texture* é fornecida pelo grupo de visão e modelagem do MIT Media Lab ([PICARD et al., 1995](#)). Originalmente, são 54 imagens de 512×512 que foram recortadas em 16 amostras de 128×128 pixels sem sobreposição, como pode ser visto na [Figura 63](#). Assim, a base é composta por um total de 864 imagens.
- *MBT*: o conjunto de dados *Multi-Band Texture* ([ABDELMOUNAIME; DONG-CHEN, 2013](#)) consiste em 154 imagens coloridas formadas pela combinação de efeitos de variações espaciais no mesmo canal e entre os canais RGB, o que é comum em imagens de alta resolução tais como de astronomia e sensoriamento remoto. No total essa base tem 2464 amostras, com 16 exemplos de 160×160 pixels não sobrepostos por classe. Algumas amostras são apresentadas na [Figura 64](#).

Figura 60 – Exemplos de imagens de textura de várias classes da base de dados Brodatz.

Fonte: [Brodatz \(1966\)](#).

Figura 61 – Amostras de textura de diferentes classes da base de dados Outex.

Fonte: [Ojala et al. \(2002\)](#).

A.1.2 Texturas Dinâmicas

Para avaliar os métodos de texturas dinâmicas, duas bases de dados referências foram consideradas:

- Dyntex++ ([GHANEM; AHUJA, 2010](#)): essa base de dados, que é uma compilação derivada da Dyntex ([PÉTERI; FAZEKAS; HUISKES, 2010](#)), tem 3600 vídeos divididos em 36 classes, sendo 100 vídeos por classe, cada um com tamanho de $50 \times 50 \times 50$ pixels. Ela é pré-processada para evidenciar somente uma única textura dinâmica por vídeo. A [Figura 66](#) mostra exemplos do primeiro quadro de amostras da Dyntex++.
- UCLA-50 ([SAISAN et al., 2001](#)): essa base de dados é composta por 50 classes, com 4 vídeos por classe, cada um com tamanho $75 \times 48 \times 48$ pixels. Nos experimentos, são utilizadas duas variações da UCLA-50, ambas propostas em ([RAVICHANDRAN; CHAUDHRY; VIDAL, 2009](#)). A primeira (UCLA-9) combina vídeos capturados de diferentes pontos de vista e os agrupam em 9 categorias: fumaça (4 amostras), flores (12), água fervente (8), mar (12), fogo (8), água (12), fontes (20), cachoeira (16) e plantas (108).

Figura 62 – Imagens de textura coloridas de 30 classes da base de dados USPTex.

Fonte: Backes, Casanova e Bruno (2012).

A segunda (UCLA-8) descarta a classe "plantas" que possui um maior número de amostras quando comparada às demais. O primeiro quadro de algumas amostras do conjunto de dados UCLA é apresentado na [Figura 65](#).

A.1.3 Formas

As quatro bases de formas utilizadas para avaliar o desempenho dos métodos de análise de formas são descritas a seguir.

Kimia

Esta base é composta de 99 imagens de contorno de formas classificadas em 9 classes com 11 formas cada ([SEBASTIAN; KLEIN; KIMIA, 2004](#)). Como pode ser observado na [Figura 67](#), a base contém formas genéricas como coelhos, homens, aviões, ferramentas, peixes etc. No conjunto, as imagens foram obtidas através de projeções de formas 3D, que podem

Figura 63 – Amostras de textura coloridas de classes da base Vistex.

Fonte: Picard *et al.* (1995).

causar a oclusão de partes do mesmo. Além disso, algumas imagens sofrem com problemas de segmentação e rotação, adicionando um desafio à análise do objeto.

ETH-80

A ETH-80 é um *benchmark* bem conhecido de formas que contém 3280 imagens separadas em 8 grupos (LEIBE; SCHIELE, 2003). Originalmente em 3D, os autores também fornecem os contornos das formas (usadas nos experimentos). As categorias são representadas por objetos feitos pelo homem e naturais e, para cada tipo de item, estilos diferentes são usados. Por exemplo, a classe xícaras, apresenta xícaras pequenas e grandes, enquanto para carros estão disponíveis diferentes marcas de carros de brinquedo. Além disso, para cada objeto, 41 imagens foram tiradas em diferentes pontos de vista, incluindo vista superior. Essa configuração adiciona variação na rotação e o contorno final pode ser muito diferente para cada tipo de objeto, aumentando a dificuldade de identificação. Esses aspectos da base pode ser visto nos exemplos da Figura 68.

Figura 64 – Exemplos de textura de diversas classes da base MBT.

Fonte: [Abdelmounaime e Dong-Chen \(2013\)](#).

Folhas Suecas

Composto de folhas de árvores da Suécia divididas em 15 espécies com 75 amostras por espécie e capturadas por um scanner de 300 dpi. Segundo os autores em [Söderkvist \(2001\)](#), as imagens não foram criadas para tornar a análise simples devido à grande variação de folhas. Uma segmentação de acordo com a diferença de cor do fundo e da folha foi necessária para obter informações de contorno, como em todos os conjuntos de dados naturais.

Folhas USP

Esta base de dados criada por pesquisadores da USP ([BACKES; CASANOVA; BRUNO, 2009a](#)) contém 600 imagens de 30 espécies de folhas. Como diferentes espécies de plantas podem conter formas semelhantes, a classificação dessa base é uma tarefa desafiadora. Além disso, no procedimento de digitalização das imagens, podem ocorrer sobreposições na adjacência dos objetos. Como outros conjuntos de dados naturais deste trabalho, essa base possui alta variação intra-classe. A [Figura 69](#) apresenta algumas amostras de diferentes classes da base de dados.

Figura 65 – Primeiro quadro de amostras de textura dinâmica de 30 classes da base de dados Dyntex++.

Fonte: Péteri, Fazekas e Huiskes (2010), Ghanem e Ahuja (2010).

A.2 Reconhecimento de Padrões

Nesta seção, são apresentados os fundamentos teóricos e conceitos dos métodos e medidas de reconhecimento de padrões empregadas neste trabalho. A tarefa de reconhecimento de padrões consiste em atribuir uma categoria ou classe y a um determinado objeto z (e.g., imagem, vídeo, rede complexa, etc) desconhecido baseado em dados de treinamento compostos pela dupla $x = (v_i, y_i)$ em que v_i é o vetor de atributos e y_i a sua respectiva classe previamente conhecida. Para realizar o processo de classificação de padrões, são empregados classificadores supervisionados com esquemas de validação. Por fim, métricas de desempenho são usadas para avaliar os modelos. Todos esses conceitos são descritos a seguir.

A.2.1 Classificadores

K-Nearest Neighbors (KNN)

O *K-Nearest Neighbors* (KNN) é uma técnica de aprendizado de máquina não paramétrica baseado em distâncias para classificação e regressão. O funcionamento desse método é bem

Figura 66 – Exemplos do primeiro quadro de amostras de textura dinâmica de diversas classes da base de dados UCLA.

Fonte: Saisan *et al.* (2001).

simples e de fácil entendimento, fazendo com que a qualidade dos vetores de características utilizados seja evidenciada. Dado uma amostra de teste y , o KNN calcula a distância de y para todas as amostras do conjunto de treinamento X para obter uma lista de distâncias de y para X . Em seguida, os k vizinhos (amostras) mais próximos do conjunto de treinamento são selecionados para realizar a predição baseado na classe majoritária. No caso de $k = 1$ (1NN) utilizado neste trabalho, a classe do vizinho mais próximo é atribuída para a amostra de teste. Exemplos de métricas de distância que podem ser utilizadas são: Euclidiana (usado nos experimentos deste trabalho), Manhattan, Hamming, Markowski etc. Observe que este método não gera um modelo, portanto, para a predição de cada instância, é necessário calcular a distância entre o teste e todas as amostras de treinamento. Esse fato torna o método significativamente caro à medida que o número de dados de treinamento usados aumenta.

Figura 67 – Amostas das 9 classes de forma da base de Kimia.

Fonte: [Sebastian, Klein e Kimia \(2004\)](#).

Linear Discriminant Analysis (LDA)

O LDA ([EVERITT; DUNN, 2001](#)) é uma técnica estatística multivariada bem estabelecida cujo conceito é a combinação linear de atributos para tarefas de regressão, classificação, etc. Esta técnica emprega uma transformação no espaço de características para obter características não correlacionadas baseadas em combinações lineares que descrevem as observações ([CASA-NOVA, 2013](#)). Na classificação, tal transformação visa projetar as características em um espaço de baixa dimensionalidade de modo que variância entre as classes seja superior que a variância das amostras da mesma classe ([EVERITT; DUNN, 2001](#)).

Considere uma variável aleatória X que pertence a uma classe $i = 1, \dots, C$ e uma função de densidade de probabilidade $f_i(x)$. O objetivo é separar o espaço de características em C diferentes regiões representando todas classes. Assim, uma nova amostra x é classificada como da classe k quando x está na região referente a classe k ([BACKES, 2010](#)). Essa predição é dada através do cálculo da probabilidade $f_i(x)$ da amostra x pertencer a cada classe i e, em seguida, atribuindo a classe de maior probabilidade $k = \arg \max_{i=1, \dots, C} f_i(x)$ para a amostra x . A função de

Figura 68 – Exemplos de cada classe de forma da base de dados ETH-80.

Fonte: [Leibe e Schiele \(2003\)](#).

probabilidade é dada por:

$$f_i(x) = -\frac{1}{2} \ln(|\Sigma|) - \frac{1}{2} (x - \mu_i)' \Sigma^{-1} (x - \mu_i), \quad (\text{A.1})$$

em que Σ é a matriz de covariância e μ_i é a média da classe i .

Support Vector Machine (SVM)

O algoritmo máquina de vetores de suporte ou *Support Vector Machine* (SVM) ([HEARST et al., 1998](#)) é um dos mais conhecidos métodos supervisionados de classificação e regressão, fornecendo bons resultados mesmo em dados com alta dimensão ([GONCALVES, 2013](#)). Nesta técnica, o objetivo é determinar um conjunto de hiperplanos de margem máxima em espaço de alta dimensão para separar as classes de dados de treinamento. Assim, o método aprende as fronteiras de decisão representadas pelos hiperplanos usando um subconjunto de vetores de treinamento ou vetores de suporte. No caso do SVM com kernel linear, usado nesta tese, a fronteira de decisão é determinada por $w \cdot x_i + b = 0$, sendo w e b os parâmetros que representam a fronteira de decisão e estimados na fase de treinamento. A classe y_i do vetor x_i é dada como:

$$y_i = \begin{cases} 1, & \text{se } w \cdot x_i + b > 0 \\ -1, & \text{se } w \cdot x_i + b < 0 \end{cases} \quad (\text{A.2})$$

No procedimento de treinamento do SVM, os parâmetros w e b são estimados visando maximizar a fronteira de decisão, o que significa minimizar a função objetivo,

Figura 69 – Amostras de contorno de folhas da base de dados Folhas USP.

Fonte: [Backes, Casanova e Bruno \(2009a\)](#).

$$\min_w = \frac{|w|^2}{2}. \quad (\text{A.3})$$

O método de multiplicadores de Lagrange pode ser empregado para resolver a minimização e estimar os parâmetros w e b , com uma transformação kernel dos vetores x_i para transformar espaços não lineares para espaços linearmente separáveis ([GONCALVES, 2013](#)).

A.2.2 Esquemas de validação

Para trabalhar com técnicas de aprendizado supervisionado é preciso utilizar conjuntos de dados de treinamento, validação e teste para avaliar os modelos. Para isso, esquemas de

validação estatisticamente confiáveis podem ser empregados para dividir o conjunto de dados em subconjuntos.

Validação Cruzada

Na estratégia de validação cruzada (em inglês, *cross-validation*) o conjunto de dados é dividido em k partições ou pastas com o mesmo número de amostras em cada partição. Essa divisão é realizada de maneira estratificada, isto é, cada partição possui a mesma distribuição de classes dos dados originais. Na validação, 1 partição é usada para testar, enquanto as $(k - 1)$ restantes são empregadas no treinamento do modelo. Esse procedimento é repetido k vezes de modo que cada partição seja usada uma vez no teste. Um caso específico utilizado neste trabalho é o *leave-one-out*. Nessa configuração, k é igual ao número de amostras do conjunto de dados, avaliando a capacidade de generalização do modelo usando todas as amostras no teste individualmente. Por fim, a acurácia final é determinada pela média das acurácias obtidas para cada uma das k partições.

Hold-out

Este esquema consiste em separar o conjunto de dados em um conjunto de treino e outro de teste baseado em uma porcentagem p . Uma divisão comum é usar 80% dos dados para treinamento e os 20% restantes para testar o modelo. Como essa divisão é aleatória, esse procedimento é repetido N vezes para garantir que os subconjuntos não favoreçam o modelo avaliado. A acurácia média e desvio padrão são reportados das acurácias calculadas para as N repetições.

A.2.3 Métricas

Para avaliar o desempenho das abordagens foram utilizados as métricas: acurácia e desvio padrão. A acurácia A é uma medida simples que pode ser calculada da matriz de confusão e, basicamente, é dada por:

$$A = \frac{N_c}{N}, \quad (\text{A.4})$$

em que N_c é o número de amostras classificadas corretamente e N o número total de amostras usadas no teste. Já o desvio padrão σ é usado como uma medida de dispersão em torno da média de N acurácias. Assim, o desvio padrão é dado por:

$$\sigma = \sqrt{\frac{\sum (A_i - \bar{A})^2}{N}}. \quad (\text{A.5})$$