



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Ana Catarina Marques Vicentim

Sopa de redes neurais randomizadas com aplicações em diagnóstico de doenças

São José do Rio Preto

2024

Ana Catarina Marques Vicentim

Sopa de redes neurais randomizadas com aplicações em diagnóstico de doenças

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Universidade Estadual Paulista "Júlio de Mesquita Filho" - (UNESP)

Instituto de Biociências, Letras e Ciências Exatas - (IBILCE)

Departamento de Ciências de Computação e Estatística

Bacharelado em Ciência da Computação

Orientador: Lucas Correia Ribas

São José do Rio Preto

2024

V633s

Vicentim, Ana Catarina Marques

Sopa de redes neurais randomizadas com aplicações em diagnóstico de doenças / Ana Catarina Marques Vicentim. -- São José do Rio Preto, 2024

51 f. : il., tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientador: Lucas Correia Ribas

1. Inteligência artificial. 2. Visão computacional. 3. Textura. 4. Redes neurais artificiais. 5. Ensemble. I. Título.

Ana Catarina Marques Vicentim

Sopa de redes neurais randomizadas com aplicações em diagnóstico de doenças

Monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Banca Examinadora

Prof. Dr. Lucas Correia Ribas

Universidade Estadual Paulista - Câmpus de São José do Rio

Preto

Orientador

Prof. Dr. Wallace Correa de Oliveira Casaca

Universidade Estadual Paulista - Câmpus de São José do Rio

Preto

Prof. Dr. Diego Renan Bruno

Universidade Estadual Paulista - Câmpus de São José do Rio

Preto

São José do Rio Preto

2024

Aos meus pais, Nelson (*in memoriam*) e Valéria, que sob muito sol, fizeram-me chegar até aqui, na sombra.

Agradecimentos

A Deus, que até aqui tem me sustentado.

À minha mãe, Valéria, minha amiga querida, você me inspira em todos os seus passos. Você é fonte de força, gentileza e também exemplo nos estudos, mesmo quando as circunstâncias tornavam difícil continuar. Guardo com carinho seu amor e apoio incondicionais, fundamentais nessa jornada.

Ao meu pai, Nelson, pelos exemplos positivos. Sempre admirei sua paixão pela leitura e escrita, e isso germinou em mim. Sem sua fundação não haveria a estrutura que sou hoje.

À minha irmã, cujo compromisso e disciplina são modelos que busco seguir. Sua determinação me inspira a trilhar meu caminho com a mesma força. Tata, quero ser como você quando eu crescer!

À tia Cidinha, pelo carinho em toda minha infância, por comemorar cada conquista minha como se fosse dela, acreditando sempre no meu potencial, mesmo quando eu ainda era só um esboço.

À minha família, minha rede de apoio que sempre me ampara, aconselha e não me deixa cair. Ao meu gatinho Quixote, e aos meus cachorrinhos Jojo, Lady e Flocos, companheiros sapecas, pelos momentos de leveza e companhia.

Aos meus amigos, os de perto e os de longe, essenciais no incentivo, que acreditaram quando eu não acreditava, que me acalmaram quando eu me desesperava. "Pros parceiros, tenho a oferecer minha presença, talvez até confusa, mas real e intensa".

À minha psicóloga, Juliana, é um privilégio te ter na minha vida. Com você aprendi o que fazer quando "sozinho, agora sou meu inimigo íntimo".

A todos os professores que contribuíram para a minha educação, e a todos que, de alguma forma, colaboraram para a realização deste trabalho.

Ao meu orientador e professor, Lucas, exemplo de pesquisador e de amor a profissão. Agradeço pelo voto de confiança, pelos ensinamentos e pela paciência. O que você me ensinou levarei para a vida toda.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pela bolsa concedida, processo nº 2022/15840-3.

À Unesp, pela oportunidade e por ter me dado tanto. Não há palavras para o que a Universidade pública pode proporcionar a um indivíduo. Há muitas entidades e

peleas queridas a quem sou grata, mas em especial agradeço a todos do restaurante universitário e da biblioteca, que juntos alimentaram corpo e alma ao longo desses quatro anos.

*“Computadores fazem arte, artistas fazem dinheiro”
(Chico Science)*

Resumo

Este trabalho aborda a análise de texturas em imagens digitais, com ênfase em suas aplicações médicas, onde o detalhamento das imagens é crucial para o diagnóstico precoce de patologias. O objetivo principal é estudar, implementar e desenvolver melhorias em técnicas de aprendizado de representações de texturas usando Redes Neurais Randomizadas (RNNs), aplicando-as no diagnóstico de doenças. Além disso, utilizou-se uma técnica de *ensemble* - a sopa de modelos - treinando diversas RNNs de maneira independente e combinando os pesos das camadas de saída para gerar representações mais robustas e precisas das texturas. O resultado esperado é a obtenção de um método de análise de textura eficiente e preciso, que supere as limitações dos métodos tradicionais, com potencial para melhorar a precisão diagnóstica em imagens médicas.

Palavras-chave: Textura; Redes Neurais Randomizadas; sopa de modelos; diagnóstico de doenças; análise de imagens.

Abstract

This work addresses the analysis of textures in digital images, with emphasis on their medical applications, where image detail is crucial for the early diagnosis of pathologies. The main objective is to study, implement and develop improvements in techniques for learning texture representations using Randomized Neural Networks (RNNs), applying them to the diagnosis of diseases. In addition, an ensemble technique - the model soup - was used, training several RNNs independently and combining the weights of the output layers to generate more robust and accurate representations of textures. The expected result is to obtain an efficient and accurate texture analysis method that overcomes the limitations of traditional methods, with the potential to improve diagnostic accuracy in medical images.

Keywords: *Texture; Randomized Neural Networks; model soup; disease diagnosis; image analysis.*

Sumário

	Lista de ilustrações	12
	Lista de tabelas	13
1	INTRODUÇÃO	16
1.0.1	Objetivos	18
2	FUNDAMENTAÇÃO E REVISÃO DA LITERATURA	19
2.1	Redes Neurais Artificiais	19
2.1.1	Redes Neurais Randomizadas	21
2.2	Ensemble de modelos	22
2.2.1	Sopa de modelos	25
2.3	Trabalhos Relacionados	26
3	MATERIAIS E MÉTODOS	30
3.1	Base de dados	31
3.2	Classificadores	32
3.3	Esquemas de validação	33
3.4	Métricas de desempenho	34
3.4.1	Acurácia	34
3.4.2	Desvio padrão	34
3.4.3	Matriz de confusão	34
3.4.3.1	Precisão	35
3.4.3.2	Sensibilidade	35
3.4.3.3	Especificidade	35
4	ABORDAGEM PROPOSTA	37
4.1	Entrada das redes	38
4.2	Transformação da imagem em vetores de padrões	38
4.3	Treinamento das RNNs	39
4.4	Representações	40
5	RESULTADOS	43
5.1	Avaliação de parâmetros	43
5.2	Comparações com a literatura	44
5.3	Diagnóstico médico de osteoartrite	46

6	CONCLUSÃO	49
	REFERÊNCIAS	51

Lista de ilustrações

Figura 2.1.1 Modelo de um neurônio	20
Figura 2.2.1 Fluxo do <i>Ensemble Learning</i>	23
Figura 3.0.1 Etapas do trabalho	30
Figura 3.3.1 Validação cruzada	33
Figura 4.0.1 Ilustração das etapas da metodologia de aprendizado de características texturas via RNNs	38
Figura 4.4.1 Sopa de modelos para aprendizado de representações de textura .	42
Figura 5.3.1 Exemplos de imagens de joelho sem OA (primeira linha) e com OA mínima (segunda linha).	47

Lista de tabelas

Tabela 1 – Matriz de confusão	35
Tabela 2 – Média das acurácias das bases Outex, Brodatz e Usptex, ao variar as medidas e Q , utilizando 5 redes neurais, e raio [3, 5, 7]	44
Tabela 3 – Acurácias das bases Outex, Brodatz e Usptex, e a média entre elas, ao variar Q e suas combinações até dois valores, fixando raio [3, 5, 7] e $m = 10$. A linha enfatizada indica a representação da textura com maior precisão média.	44
Tabela 4 – Acurácias das bases Outex, Brodatz e Usptex, ao variar a quantidade de redes neurais m , utilizando $Q = \{9, 29\}$ e raio [3,5,7]	45
Tabela 5 – Comparação de acurácias de diferentes métodos de análise de textura em três bancos de dados de texturas.	46
Tabela 6 – Acurácia usando o classificador LDA na base de dados 1200Tex.	46
Tabela 7 – Acurácias de classificação usando SVM para os métodos aplicados na base de OA.	48

Lista de algoritmos

1 Pseudo-código para treinamento de uma RNN com uma camada oculta, em que Q é o número de neurônios ocultos.232 Sopa gulosa26

Lista de abreviaturas e siglas

CNN	<i>Convolutional Neural Networks</i>
KNN	<i>K-Nearest Neighbours</i>
LBP	<i>Local Binary Pattern</i>
LCG	<i>Linear Congruent Generator</i>
LDA	<i>Linear Discriminant Analysis</i>
RNA	<i>Redes Neuraís Artificiais</i>
RNNs	<i>Randomized Neural Networks</i>
SVM	<i>Support Vector Machine</i>
ViTs	<i>Vision Transformers</i>

1 Introdução

Textura é uma propriedade visual fundamental presente em muitos objetos, superfícies de materiais, na natureza, em imagens do dia-a-dia, imagens de microscopia e médica, entre outras. Em imagens digitais, os padrões de texturas são formados pela variação e organização de pixels, resultando em padrões contraditórios que vão desde regulares até complexos que estão relacionados diretamente com a superfície representada. De fato, não existe um consenso na comunidade científica para uma definição formal e única de texturas, pois esta é motivada muitas vezes pela percepção individual ou problema estudado (RIBAS, 2017). Ao longo das décadas, a computação tem se tornado parte essencial da medicina, haja vista que o desenvolvimento científico dessa área elevou a qualidade e a precisão do diagnóstico e do tratamento de doenças. Graças as modernas técnicas de obtenção de imagens no meio médico, hoje é possível obter imagens com alto grau de detalhamento, possibilitando a detecção precoce de patologias e auxiliando o profissional da saúde na tomada de decisões.

Essa diversidade de definições e aplicações resulta em uma extensa e heterogênea literatura de métodos que exploram diferentes paradigmas para descrição de texturas e são empregados em diversas áreas de modo a beneficiar sistemas de reconhecimento de imagens. Apesar da diversidade de métodos, muitos aspectos da textura são difíceis de modelar e analisar, mesmo em cenários de imagens controladas. Além disso, a natureza selvagem (não-linear) das imagens digitais também resulta em variabilidade adicional, tornando a tarefa ainda mais complexa em aplicações do mundo real. Como levantado por Liu (LIU et al., 2019), apesar de décadas de investigação, a maioria dos métodos de análise de texturas ainda não conseguem lidar de forma suficiente com texturas do mundo real e são computacionalmente complexos, sendo o principal desafio equilibrar descritores de texturas de alta qualidade e eficientes computacionalmente.

Ao contrário das abordagens baseadas em complexas arquiteturas que demandam alto custo (como as *Convolutional Neural Networks* (CNNs) e *Vision Transformers* (ViTs) - (BAHROUN; ABED; ZAGROUBA, 2023; MELEKODAPPATTU et al., 2023)), difícil interpretação e muitas vezes grande quantidade de dados (em casos de treinamento), as RNNs proporcionam descritores de texturas baseados em um paradigma de aprendizado de representação simples, de baixo custo computacional e que utiliza poucos dados. Essa simplicidade se deve graças ao treinamento dos pesos da camada de saída que é feito por uma solução de forma fechada (solução de mínimos quadrados), em contraste com o aprendizado baseado em gradiente descendente. Na

literatura, o trabalho pioneiro envolvendo RNNs e texturas foi desenvolvido por Sá Junior (JUNIOR; BACKES, 2016), que propôs um descritor baseado no treinamento de RNNs individuais para cada imagem usando um conjunto de vetores de padrões locais como entrada. Em seguida, outros trabalhos foram desenvolvidos para análise de imagens inspirados nessa ideia, tais como a combinação com redes complexas (RIBAS et al., 2020), LBP e a aplicação em texturas dinâmicas (vídeos) (RIBAS et al., 2022; JUNIOR; RIBAS; BRUNO, 2019) e análise de contornos (JUNIOR; BACKES; BRUNO, 2018). Os resultados promissores e vantagens deste tipo de abordagem abriram um campo de pesquisa com bastante potencial para ser explorado em análise de imagens e texturas.

As RNNs se diferenciam das técnicas tradicionais de redes neurais profundas por sua simplicidade e eficiência. Com apenas uma camada oculta, as RNNs aprendem rapidamente e apresentam alto desempenho, mesmo com conjuntos de dados de treinamento reduzidos. Essa combinação de características torna as RNNs ideais para o aprendizado de representações utilizando a técnica de sopa de modelos. Através da sopa de modelos, as RNNs podem extrair características relevantes de texturas, preservando as propriedades desejáveis de simplicidade, processamento rápido e baixo volume de dados. A sopa de modelos consiste na estratégia proposta por Wortsman, et al. (WORTSMAN et al., 2022). Nesta abordagem, ao invés de otimizar os pesos de uma única rede neural, opta-se por treinar diversas redes neurais de maneira independente. Posteriormente, os pesos resultantes desses modelos treinados individualmente são combinados por meio da média, gerando um modelo final mais robusto. Inspirado nesse método, a ideia proposta aqui consiste em treinar cada RNN com conjuntos de pesos iniciais aleatórios distintos na camada de entrada, combinando posteriormente os pesos treinados da camada de saída por meio de métodos estatísticos. Como os pesos aleatórios da camada de entrada da RNN tem como função projetar os dados de entrada em um outro espaço dimensional, essa abordagem busca aproveitar-se de diferentes projeções para criar uma representação final mais robusta.

Esse método sustenta-se nas técnicas de *Ensemble*, que se beneficiam da diversidade de vários modelos combinados para criar soluções mais robustas e eficazes, em contrapartida as técnicas clássicas em aprendizado de máquina, onde há utilização de um único modelo que sofre ajustes de seus diversos parâmetros. Assim, *Ensemble Learning* é uma técnica de aprendizado de máquina que envolve a combinação de diversos modelos base para criar um modelo consolidado, que é mais robusto e poderoso do que seus componentes individuais. Essencialmente, um conjunto de classificadores é treinado com dados de entrada, produzindo previsões que são, posteriormente, combinadas para formar uma previsão final. Para chegar a essa previsão final, são empregadas diferentes técnicas, como a de votação média, na qual

as saídas de cada modelo são calculadas e a média entre elas determina a saída do modelo agregado.

A sopa de RNNs consiste na agregação de pesos treinados de diversas RNNs, cada uma inicializada com pesos aleatórios diferentes na camada de entrada. Através dessa técnica, busca-se gerar representações mais robustas e precisas das texturas, superando o desempenho de uma única RNN e de outros métodos da literatura. Para a agregação dos pesos, serão exploradas diferentes medidas, como a média, o desvio padrão e a soma. Cada uma dessas medidas apresenta características únicas que podem influenciar a qualidade das representações finais. A escolha da medida ideal será determinada empiricamente, levando em consideração fatores como a acurácia, a sensibilidade, a especificidade e a quantidade de *features*.

1.0.1 Objetivos

O objetivo principal é estudar, implementar e desenvolver melhorias em técnicas de aprendizado de representações de texturas usando Redes Neurais Randomizadas (RNNs). Também busca-se a aplicação direta dos métodos estudados e desenvolvidos no diagnóstico de doenças. Mais especificamente, este trabalho tem os seguintes objetivos:

- (i) Investigar e identificar limitações e possíveis melhorias nas abordagens pré-existentes da literatura para caracterização de texturas com RNNs.
- (ii) Propor aprimoramentos e desenvolver abordagens para aprendizado de representações usando RNNs visando entregar métodos de análise que equilibrem a extração discriminativas representações de texturas, baixa simplicidade computacional e que usam poucos dados.
- (iii) Testar experimentalmente a sopa de modelos de RNNs, identificando, as melhores combinações de parâmetros para obter representações de texturas discriminativas.
- (iv) Aplicar os métodos de análise de textura estudados em imagens visando o diagnóstico de doenças.

2 Fundamentação e Revisão da Literatura

Em seguida se segue a fundamentação e revisão da literatura, abordando conceitos-chave no campo da inteligência artificial e aprendizado de máquina. Primeiramente, serão exploradas as redes neurais, que são modelos computacionais inspirados no funcionamento do cérebro humano, amplamente utilizados para resolver problemas complexos em diversas áreas. Em seguida, são discutidas as redes neurais aleatórias, uma variação das redes neurais tradicionais que utiliza conexões estabelecidas de maneira aleatória para melhorar a eficiência e desempenho em certas aplicações. Também é abordado o conceito de *ensemble* de modelos, uma técnica que combina múltiplos modelos de aprendizado para obter melhores resultados e maior robustez. Por fim, é examinada a sopa de modelos, uma abordagem inovadora que integra diversos modelos para criar uma solução ainda mais poderosa e precisa, aproveitando a diversidade e complementaridade dos modelos envolvidos.

2.1 Redes Neurais Artificiais

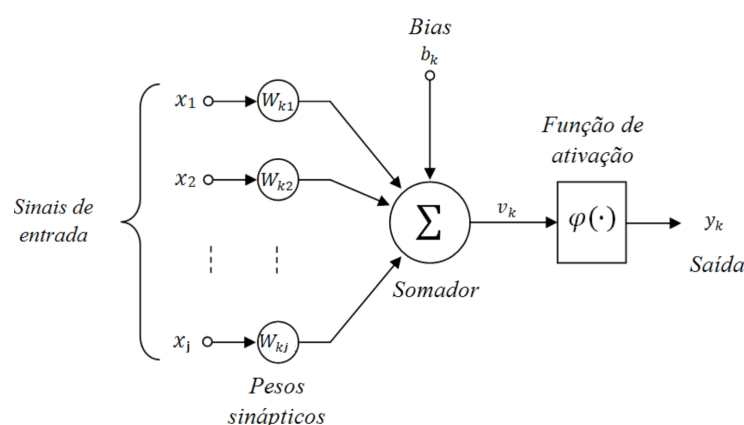
Como Haykin descreve em seu livro (HAYKIN, 2009), as redes neurais artificiais (RNAs) são sistemas computacionais inspirados no funcionamento do cérebro humano. Compostas por unidades interconectadas chamadas "neurônios", as RNAs aprendem e se adaptam a partir de dados, imitando o processo de aquisição de conhecimento e armazenamento de memórias no cérebro.

Ao contrário da programação tradicional, onde as instruções são definidas manualmente, as RNAs aprendem através de um processo chamado "aprendizagem". Nesta etapa, a rede é exposta a um conjunto de dados e ajusta os pesos de suas "sinapses" - as conexões entre os neurônios - para minimizar um erro ou otimizar um desempenho. Esse processo iterativo permite que a rede aprenda padrões complexos e generalize seu conhecimento para novos dados.

A modificação dos pesos sinápticos é o método tradicional para treinar redes neurais, se baseando na atualização iterativa dos pesos para minimizar um erro ou otimizar um objetivo específico. Além disso, as RNAs também podem modificar sua própria topologia, criando ou removendo conexões entre os neurônios. Essa capacidade de adaptação estrutural é motivada pela observação de que o cérebro humano também apresenta a morte de neurônios e o crescimento de novas conexões sinápticas ao longo da vida.

O neurônio artificial é a unidade básica de processamento em uma rede neural, inspirado no funcionamento do neurônio biológico. Cada neurônio recebe diversos sinais de entrada, cada um ponderado por um peso sináptico associado à respectiva conexão. Esses pesos representam quão importante é a conexão entre aqueles neurônios e podem ser representados por valores positivos, negativos ou nulos. O neurônio artificial possui três elementos principais, sendo os pesos, a função somadora e a função de ativação. Cada entrada é recebida através de uma sinapse, com um peso associado. Tendo esses valores, a soma ponderada das entradas é realizada por um somador, gerando a ativação bruta do neurônio. Então, a função de ativação age limitando a amplitude do sinal de saída, introduzindo não linearidade no modelo. Essa representação pode ser entendida na Figura 2.1.1.

Figura 2.1.1 – Modelo de um neurônio



Fonte: Adaptado de (HAYKIN, 2009)

O modelo neural apresentado na Figura 2.1.1 incorpora um elemento adicional crucial: o viés, ou *bias*, representado por b_k . Esse viés atua como um deslocamento constante aplicado à entrada do neurônio antes da função de ativação. O viés é adicionado a soma ponderada dos valores de entrada na equação de ativação do neurônio k e pode ser positivo ou negativo, influenciando diretamente a ativação do neurônio, pois um viés positivo aumenta a ativação, enquanto um viés negativo a diminui. A existência do *bias* facilita o processo de treinamento da rede neural, acelerando a convergência para uma solução ideal.

Matematicamente, o neurônio k pode ser descrito pelas equações 2.1.1 e 2.1.2.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.1.1)$$

$$y_k = \varphi(u_k + b_k) \quad (2.1.2)$$

onde $x_1, x_2, x_3, \dots, x_m$ são os sinais de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos do neurônio k ; u_k é a saída correspondente a uma determinada entrada que passou pelo combinador linear; b_k é o viés; $\varphi()$ é a função de ativação; e y_k é o sinal de saída do neurônio.

Em rede neurais os neurônios estão dispostos em camadas. Na forma mais básica de uma rede, existe uma camada inicial de nós de entrada que se conecta diretamente a uma camada final de neurônios, o que é conhecido como uma rede *feedforward*, que possui camada única e não possui retorno de sinal, uma vez que ela apenas “caminha para frente”.

As redes neurais *feedforward* podem ser mais complexas, contendo uma ou mais camadas ocultas - porque elas não são visíveis nas entradas ou saídas. Os neurônios, que também serão denominados ocultos, mediam a entrada e a saída da rede de maneira eficaz. Com mais camadas, essa rede possui a capacidade de identificar padrões mais complexos, pois fará mais interações neurais.

Na camada de entrada da rede, os nós de origem fornecem o vetor de entrada, que são os sinais aplicados aos neurônios da segunda camada (a primeira camada oculta). Os sinais de saída dessa segunda camada são então usados como entradas para a terceira camada, e assim por diante, até a última camada da rede. Na maioria das redes, os nós recebem como entrada os sinais de saída da camada anterior. Os sinais de saída da última camada serão a resposta aos valores determinados pelos nós da camada de entrada.

2.1.1 Redes Neurais Randomizadas

A ideia deste trabalho é utilizar os pesos obtidos na saída da RNN para descrever imagens. É importante ressaltar que vários experimentos aqui realizados foram inspirados na abordagem de Sá Junior et al. (JUNIOR; BACKES, 2016). Inicialmente, a RNN começa com pesos aleatórios entre a camada de entrada e a camada oculta, representados por uma matriz W , e sua saída é determinada pelo método dos mínimos quadrados. Na matriz de pesos W , os valores influenciam diretamente no resultado do treinamento da rede neural e, conseqüentemente, nos pesos treinados da camada de saída (que serão as *features* usadas para representar as imagens). Assim, é necessário usar os mesmos valores na matriz de pesos W para garantir que uma imagem específica, quando processada com os mesmos parâmetros, sempre produza um vetor de características único. Para isso, o *Linear Congruent Generator (LCG)* (PARK; MILLER, 1988) é utilizado para gerar os números pseudo-aleatórios que compõem a matriz W , utilizando a equação 2.1.3.

$$V(n + 1) = (a * V(n) + b) \bmod c, \quad (2.1.3)$$

onde V é uma sequência aleatória de comprimento $E = Q * (p + 1)$ e $V(1) = E + 1$. Os valores dos parâmetros a , b e c são definidos como $a = E + 2$, $b = E + 3$ e $c = E^2$. Para construir a matriz W , o vetor V é dividido em Q segmentos de comprimento $p + 1$. A matriz W é normalizada com média zero e desvio padrão unitário.

Em seguida, considera-se a matriz de características de entrada X com todos os valores das vizinhanças dos pixels, e um vetor D com os valores dos pixels centrais correspondentes a cada vizinhança. Após cada valor de X ter sido conectado com o *bias* b - que representa quantitativamente as chances do neurônio ser ativado - multiplicamos as matrizes W e X e, em seguida, aplicamos uma função de ativação do neurônio, resultando em uma nova matriz Z . Essa matriz, assim como X , é uma matriz de características da amostra. No processo de treinamento, os pesos da camada de saída são organizados em uma matriz M (que será um vetor caso haja apenas um neurônio na camada de saída). O objetivo da RNN é encontrar a matriz M que, ao ser multiplicada por Z , resulte na saída desejada D , ou seja, determinar os pesos de saída da rede neural que mapeiam corretamente a projeção aleatória das características de entrada na saída desejada.

Em termos de caracterização de imagens, esse processo permite que a RNN aprenda os pesos de saída que preveem a intensidade de um pixel com base em sua vizinhança, matematicamente representada pela equação 2.1.4.

$$M = DZ^T (ZZ^T + \lambda I)^{-1}, \quad (2.1.4)$$

onde $\lambda > 0$ e I é a matriz identidade. Assim, o vetor de características que representa a imagem de textura é construído usando M . Este é um vetor unidimensional, visto que a rede é treinada com um único neurônio de saída correspondente ao valor do pixel central. Vale notar que diversos vetores M podem ser combinados para formar o vetor final, levando em conta os parâmetros Q (número de neurônios ocultos) e o tamanho da janela.

O algoritmo 1 descreve os passos em pseudo-código da função para treinamento de uma RNN.

2.2 Ensemble de modelos

Em aprendizado de máquina, é comum a utilização de um único modelo que sofre ajustes de seus diversos parâmetros, afim de calibrá-los para atingir os melhores

Algoritmo 1 Pseudo-código para treinamento de uma RNN com uma camada oculta, em que Q é o número de neurônios ocultos.

Function trainRNN(X, D, Q):

```

X = Zscore(X) // normalizar a matriz de vetores de entrada
W = LCG(Q,P+1,Q*(P+1)) // gerar os pesos aleatórios
X = addBias(X,-1) // add o bias na matriz de entrada

Z = Activation(W*X) // função de ativação da camada oculta
Z = addBias(Z,-1) // add o bias em Z

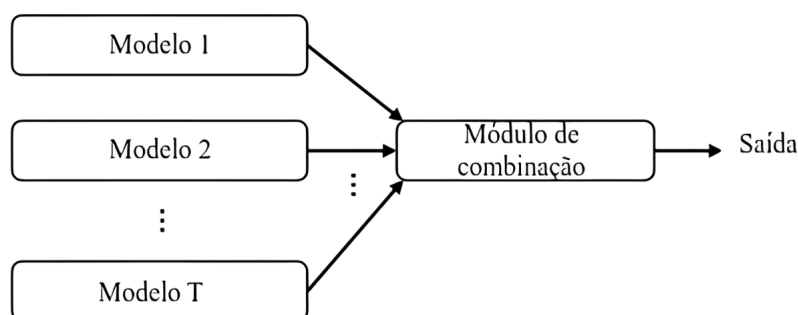
lambda = 0.001
M = (D*Z')/(Z*Z'+ lambda * eye(Q+1)) // calcula os pesos de saída com a
peudo-inversa de Moore-Penrose

return M

```

resultados. Por outro lado, há a ideia de aplicar técnicas de *Ensemble*, que se beneficiam da diversidade de vários modelos combinados para criar soluções mais robustas e eficazes. *Ensemble learning* envolve o treinamento e a combinação de vários modelos para resolver um problema de aprendizagem. Conforme ilustrado na Figura 2.2.1, o fluxo começa com o treinamento de um conjunto de modelos individuais, que são posteriormente combinados utilizando diversas estratégias. Cada modelo individual é normalmente treinado utilizando um algoritmo de aprendizagem já existente.

Figura 2.2.1 – Fluxo do *Ensemble Learning*



Fonte: Adaptado de (ZHOU; ZHOU, 2021)

Um *Ensemble* é considerado homogêneo quando todos os modelos individuais são do mesmo tipo, como por exemplo, um “*Ensemble* de rede neural” contém apenas redes neurais. Nestes casos, os modelos individuais são chamados de modelos base, e os algoritmos de aprendizagem correspondentes são conhecidos como algoritmos de aprendizagem base. Por outro lado, um *Ensemble* heterogêneo inclui diferentes tipos de modelos e algoritmos de aprendizagem individuais. Neles, os modelos individuais são chamados de modelos componentes ou simplesmente de modelos individuais.

A combinação de múltiplos modelos em um *Ensemble* geralmente resulta em uma capacidade de generalização muito superior à de qualquer modelo individual, especialmente no caso dos chamados “modelos fracos”, ou *weak learners*. Assim, embora um conjunto de modelos fracos possa teoricamente alcançar um bom desempenho, costuma-se preferir modelos fortes (ou *strong learners*) por reduzir o número de modelos individuais necessários. É notório que a combinação de modelos com diferentes qualidades tende a produzir um resultado melhor que o modelo mais fraco, mas pior que o modelo mais forte. Intuitivamente, o questionamento levantado é porque utilizar *Ensemble* em vez de apenas aplicar o melhor modelo individual.

Como (ZHOU; ZHOU, 2021) explica em seu livro, a medida que o número de modelos base no *Ensemble* aumenta, a taxa de erro diminui exponencialmente e eventualmente se aproxima de zero. Sendo assim, é perceptível que precisão e diversidade são duas propriedades cruciais para se obter modelos condizentes com a realidade. Entretanto, elas são conflitantes, pois geralmente, quando a precisão já é alta, é necessário sacrificar um pouco dessa precisão para aumentar a diversidade. Assim, a geração e a combinação de modelos individuais precisos e diversos são questões fundamentais para o *Ensemble Learning*.

Os métodos atuais de *Ensemble Learning* podem ser divididos em duas categorias, que dependem de como os modelos individuais são gerados. A primeira categoria, representada pelo *Boosting*, cria modelos individuais com fortes correlações e gera os modelos de forma sequencial. A segunda categoria, representada pelo *Bagging*, cria modelos individuais de forma independente e pode paralelizar o processo de geração.

Segundo Dietterich (DIETTERICH, 2000) a combinação de modelos individuais oferece vantagens. A primeira delas se trata da perspectiva estatística. Como existem muitas formas de se compor um modelo, é comum que vários modelos apresentem desempenhos semelhantes. Quando se escolhe um único modelo, o desempenho depende inteiramente da qualidade desse modelo específico. No entanto, ao combinar diversos modelos, o risco de selecionar um modelo de desempenho inferior é reduzido.

Outro ponto é que os algoritmos de aprendizado muitas vezes ficam presos nos chamados ótimos locais, o que pode resultar em um desempenho de generalização ruim, mesmo quando há muitos dados de treinamento disponíveis. Ao repetir o processo de aprendizado diversas vezes e combinar os modelos de maneiras diferentes, o risco de ficar preso em um ótimo local ruim é diminuído, embora não haja garantia de encontrar o ótimo global.

Uma das maneiras de combinar modelos é justamente a estratégia da sopa de modelos, descrita a seguir e que tem obtido resultados significativos.

2.2.1 Sopa de modelos

Mais recentemente, Wortsman, et al. (WORTSMAN et al., 2022) propôs a uma estratégia de *Ensemble* chamada de sopa de modelos. Nesse método, é feita uma analogia com o preparo de uma receita, onde diferentes ingredientes são adicionados e testados individualmente, visando uma composição mais adequada. Assim, cada modelo de inteligência artificial é tratado como um "ingrediente" adicionado à sopa, avaliando-se, após cada inclusão, se houve uma melhoria na precisão do modelo agregado. Caso o modelo contribua positivamente para a robustez da combinação, ele permanece; caso contrário, é substituído, visando o aperfeiçoamento do resultado final.

Nesta abordagem, ao invés de otimizar os pesos de uma única rede neural, opta-se por treinar diversas redes neurais de maneira independente. Posteriormente, os pesos resultantes desses modelos treinados individualmente são combinados por meio da média, gerando um modelo final mais robusto. Como os pesos aleatórios da camada de entrada da rede neural tem como função projetar os dados de entrada em um outro espaço dimensional, essa abordagem busca aproveitar-se de diferentes projeções para criar uma representação final mais robusta.

O processo para construção de uma sopa de modelos se inicia no treinamento independente de cada um dos modelos. Desse modo, cada uma das redes é inicializada com diferentes pesos aleatórios, e também podem ser treinadas com diferentes subconjuntos de dados. Treinando várias redes separadamente, o algoritmo explora diferentes regiões do espaço de hipóteses. Um espaço de hipóteses \mathcal{H} contém a classe de funções consideradas por um algoritmo de aprendizagem, que irá escolher a função $g \in \mathcal{H}$ baseado em uma função de erro calculada em um conjunto de treinamento. A junção destas três partes é o modelo de aprendizagem. Isso significa que cada rede pode capturar diferentes aspectos dos padrões nos dados.

Em seguida, após o treinamento, os pesos das diferentes redes neurais são combinados através da média. Então, é criado um modelo final que é a média dos pesos dos modelos individuais. A média dos pesos tende a suavizar as características individuais dos modelos, resultando em uma combinação que aproveita as vantagens de cada um. Em seu trabalho, Wortsman define o algoritmo 2 para explicar o método.

Algoritmo 2 Sopa gulosa

Input: Ingredientes candidatos para a sopa $\{\theta_1, \dots, \theta_k\}$ (em ordem decrescente de $\text{ValAcc}(\theta_i)$)
Output: Modelo médio dos ingredientes selecionados
ingredientes $\leftarrow \{\}$
for $i = 1$ **to** k **do**
 if $\text{ValAcc}(\text{média}(\text{ingredientes} \cup \{\theta_i\})) \geq \text{ValAcc}(\text{média}(\text{ingredientes}))$ **then**
 ingredientes \leftarrow ingredientes $\cup \{\theta_i\}$
 end
end
return $\text{média}(\text{ingredientes})$

onde a entrada $\{\theta_1, \dots, \theta_k\}$ é a lista de modelos candidatos que serão considerados para a combinação. Estes modelos estão ordenados em ordem decrescente de precisão na validação (ValAcc). Inicia-se com um conjunto vazio de ingredientes (modelos) que serão incluídos na combinação final. O laço `for` percorre cada modelo candidato da lista. O `if` verifica se a inclusão do modelo atual no conjunto de ingredientes melhora ou mantém a precisão na validação da média dos modelos no conjunto de ingredientes. Se a condição for satisfeita, o modelo é adicionado ao conjunto de ingredientes. Ao final, após percorrer todos os modelos, a saída é a média dos modelos no conjunto de ingredientes retornada como o modelo combinado final.

Assim, o algoritmo tenta construir um conjunto de modelos que, quando combinados, oferecem uma precisão que é muitas vezes melhor do que quando um único modelo é utilizado. Ele começa com um conjunto vazio e, iterativamente, adiciona modelos à combinação apenas se isso não piorar a precisão. A combinação final é a média dos modelos selecionados, o que tende a proporcionar um desempenho mais robusto.

2.3 Trabalhos Relacionados

A definição de textura varia amplamente na comunidade científica, muitas vezes influenciada pela percepção individual ou pelo problema em questão, portanto, é natural que na computação surgissem diferentes abordagens para descrever texturas. Essas abordagens incluem métodos clássicos que se baseiam em descrições matemáticas ou “feitas à mão” dos padrões de texturas, assim como abordagens mais recentes que exploram redes neurais artificiais. Das técnicas *handcrafted*, tem destaque as baseadas em estatísticas de primeira e segunda ordem relacionadas aos níveis de cinza dos pixels (eg., descritores de Haralick (HARALICK; SHANMUGAM; DINSTEN, 1973)); baseados no conceito do método de padrões binários locais (*Local Binary Pattern* - LBP)(HOSNY et al., 2023; LUO et al., 2023; FADAEI; HOSSEINI;

RAHIMIZADEH, 2023); estruturais, que consideram a textura como constituída por uma organização hierárquica de arranjos espaciais de micro texturas bem definida (SILVA et al., 2023); baseado em modelos que medem a complexidade/fractalidade da textura (PARADOWSKA-STOLARZ et al., 2023; GONG et al., 2023; FLORINDO, 2024); espectrais que convertem a imagem em uma nova baseado em propriedades de frequências espaciais da intensidade dos pixels (ATTALLAH, 2023; PATEL; KASHYAP, 2023).

Entretanto, apesar das técnicas *handcrafted* serem precursoras na classificação de texturas, elas enfrentam diversas dificuldades (NANNI; GHIDONI; BRAHNAM, 2017). Elas podem ser computacionalmente caras, devido aos seus cálculos detalhados, que podem impactar a velocidade de processamento. O custo computacional também pode ser elevado quando os recursos artesanais resultam em vetores de recursos de alta dimensão, causando maior complexidade computacional. Outro problema é que essas técnicas podem ter dificuldade para generalizar bem em diferentes conjuntos de dados ou cenários, potencialmente levando a menor precisão. Além disso, pode ocorrer sobreajuste, que é quando um modelo se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados. Isso pode ocorrer porque os métodos *handcrafted* são projetados especificamente para certas características, potencialmente reduzindo sua adaptabilidade a novos dados. Em vista disso, as RNNS podem ser uma alternativa, levando em consideração seu baixo custo computacional e seu alto desempenho com bases de dados pequenas, que por sua vez geram vetores de características pequenos. Além disso, as RNNS têm obtido ótimos resultados, ultrapassando os métodos tradicionais em análise de textura (JUNIOR; CORTEZ; BACKES, 2014).

Mais recentemente, foram os avanços significativos em *Deep Learning* que tornaram as redes neurais a abordagem dominante em Visão Computacional. Diversas arquiteturas de *Deep Convolutional Neural Network* foram propostas (KAUR; SINGH, 2023) para o reconhecimento de objetos, cada uma com suas próprias características estruturais, compostas por diferentes blocos convolucionais e suas combinações. Com o contínuo avanço dos algoritmos de aprendizado profundo houve uma melhora significativa no desempenho desses modelos. No entanto, isso não significa que os métodos tradicionais, desenvolvidos ao longo de décadas antes do surgimento do aprendizado profundo, tenham se tornado obsoletos. Existem situações em que os métodos convencionais, utilizando características globais, ainda são a melhor escolha.

Sendo assim, trabalhos que utilizam *Deep Learning* continuam enfrentando diversos desafios, como por exemplo questões de eficiência e escalabilidade. Essa questão é exposta quando com o aumento do número de classes de objetos, a com-

plexidade computacional também cresce, demandando mais recursos de computação devido ao grande número de regiões dentro de uma única imagem. Outro problema enfrentado em arquiteturas desse tipo é o desequilíbrio de classe, quando ocorre a distribuição desigual de dados entre classes. Basicamente, isso ocorre quando uma classe contém um número desproporcional de instâncias em comparação com outras classes. Geralmente, para se resolver esse problema são gerados dados sintéticos. Nesse sentido, as RNNs apresentam vantagens em relação aos métodos de *Deep Learning*. Elas são mais eficientes em termos computacionais, uma vez que não requerem treinamento extensivo. Além disso, não é necessário gerar dados sintéticos ao utilizar as RNNs, pois suas decisões são baseadas em projeções globais dos dados. Isso as torna muito úteis em situações onde os recursos são limitados ou quando se trabalha com conjuntos de dados desequilibrados.

Ademais, outras técnicas baseadas em redes neurais artificiais têm sido exploradas para análise de texturas, alcançando resultados promissores (TUSSUPOV et al., 2023; GHAHFARROKHI et al., 2023). Além das técnicas baseadas em *Convolutional Neural Networks* (CNNs) e *Vision Transformers* (ViTs) (SCABINI et al., 2024) que têm alcançado bons resultados em análise de textura se aproveitando de grandes arquiteturas treinadas com enormes bases de dados de imagens, as Redes Neurais Randomizadas (*Randomized Neural Networks* - RNNs) também tem se mostrado promissoras (FARES; RIBAS, 2024; RIBAS et al., 2024b).

Além disso, as RNNs também obtiveram bom desempenho para analisar conjuntos de dados compostos por poucas imagens formadas por padrões complexos de texturas e cor, como é comum em imagens médicas, como mostram os trabalhos de diagnóstico de câncer (RODRIGUES et al., 2021), covid-19 (SOARES et al., 2020) e Papanicolaou (JUNIOR; BACKES; BRUNO, 2018). Em particular, os trabalhos das refs. (RODRIGUES et al., 2021; SOARES et al., 2020) são pioneiros em aplicar técnicas de análise de texturas e aprendizado de máquina na análise de imagens de unidades de biossensor, pois apesar de existirem muitos estudos que analisem imagens biológicas, poucos foram reportados usando imagens de unidades sensoriais.

Apesar dos ótimos resultados, as RNNs utilizam pesos aleatórios na camada de entrada para projetar os dados em um novo espaço dimensional. Embora essa técnica possa ser eficaz em alguns casos, ela também apresenta uma série de limitações. A projeção aleatória pode não ser ideal para capturar todas as nuances dos dados de entrada, resultando em representações subótimas. Além disso, a dependência de uma única projeção aleatória pode levar a uma maior sensibilidade a ruídos e variações nos dados, comprometendo a robustez e a generalização do modelo.

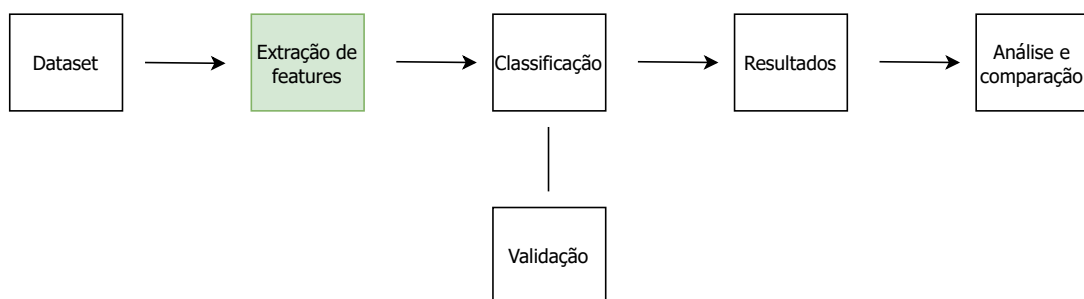
A sopa de modelos propõe uma solução para essas desvantagens ao treinar múltiplas redes neurais com diferentes conjuntos de pesos iniciais aleatórios na

camada de entrada. Isso permite explorar uma variedade de projeções dos dados, aumentando a chance de capturar informações relevantes de forma mais completa. Além disso, a combinação dos pesos das camadas de saída dos modelos individuais por meio de métodos estatísticos, como a média, contribui para uma representação final mais robusta e estável. Essa estratégia pode trazer avanços significativos para o campo do aprendizado de máquina e visão computacional, proporcionando modelos mais confiáveis e com melhor desempenho para análise de textura.

3 Materiais e métodos

A metodologia do trabalho consiste inicialmente na caracterização de texturas, posteriormente passando pela classificação e validação do resultado. A partir disso, é possível realizar experimentos para entender melhor a eficiência das redes em conjunto (com a sopa de modelos) ou individualmente. Por último, serão realizadas análises acerca dos resultados, como ilustra a 3.0.1. É importante frisar que o trabalho realizado aqui é na etapa de extração de *features*.

Figura 3.0.1 – Etapas do trabalho



Fonte: Autoria própria

Na etapa de classificação, será utilizado um método que aprenda representações de imagens e utiliza um algoritmo tradicional de aprendizado e máquina para classificá-las. As representações computadas pelo método desenvolvido serão analisadas em tarefas de classificação. Nele, as representações e rótulos serão usados para treinar um algoritmo de classificação estabelecido para criar modelos de predição.

Com o *framework* para criação das RNNs estabelecido, inicia-se as experimentações com a sopa de modelos. Inicialmente, as imagens serão normalizadas e preparadas para o treinamento das RNNs. Isso incluirá a transformação das imagens em vetores de padrões locais conforme descrito anteriormente, usando diferentes estratégias para obter representações robustas. Em seguida, um conjunto de RNNs será treinado de forma independente, cada uma com diferentes pesos iniciais aleatórios. Cada RNN aprenderá a partir dos vetores de padrões locais gerados no passo anterior. Serão exploradas diferentes configurações de hiperparâmetros, como a quantidade de neurônios da camada oculta. Ao final, os pesos aprendidos das camadas de saída das RNNs serão combinados para formar um modelo final. A abordagem de sopa de modelos será utilizada para combinar esses pesos, utilizando a média ou outras técnicas estatísticas para agregar as representações. Será investigada a melhor forma

de realizar essa combinação para otimizar a performance do modelo final.

Como últimas etapas, ocorrerá a validação e análise dos métodos. A eficácia do modelo combinado será avaliada utilizando métricas padrão de classificação, como acurácia e *recall*. Serão realizadas comparações com outros métodos da literatura, incluindo redes neurais tradicionais, RNNs de forma unitária e abordagens de *ensemble* mais comuns, para verificar a superioridade da sopa de modelos. Também serão utilizadas diversas métricas de desempenho para avaliar os resultados e a eficiência computacional dos métodos.

3.1 Base de dados

As bases de imagens são usualmente empregadas em estudos de texturas, com o objetivo de estudar as abordagens e comparar os resultados entre diferentes abordagens da literatura. Desse modo, foram utilizadas as seguintes bases nesse estudo:

- Brodatz (BRODATZ et al., 1966): essa base contém 1776 imagens de tamanho 128 por 128 pixels em níveis de cinza, separadas em 111 classes com 16 amostras.
- USPtex (BACKES; CASANOVA; BRUNO, 2012): é composta por 2160 imagens coloridas de tamanho 128 por 128 pixels constituída de texturas comuns no cotidiano, organizadas em 180 classe com 12 amostras cada.
- 1200tex (CASANOVA; JUNIOR; BRUNO, 2009): inicialmente, essa base foi composta por imagens de plantas de 20 espécies, com cada classe possuindo 20 amostras. De cada amostra foram recortadas figuras de 128 por 128 pixels, resultando em 1200 imagens, divididas em grupos de 60 imagens cada classe.
- Outex (OJALA et al., 2002): é um conjunto de 68 classes de texturas coloridas, com 54 amostras cada. As imagens foram compostas em três iluminações distintas, além de terem sido rotacionadas.
- OAI(OAI. . . , 2018-11-14): essa base foi criada pela OsteoArthritis Initiative (OAI) para realizar uma classificação entre joelhos com e sem osteoartrite, com o objetivo de avaliar a precisão do modelo desenvolvido nesse projeto. A base de dados conta com 688 imagens de joelho, sendo metade destes indivíduos saudáveis, e metade com osteoartrite.

3.2 Classificadores

De forma geral, esses métodos para análise de texturas podem ser divididos em dois tipos: (i) métodos que extraem/aprendem representações (vetores) das imagens e empregam algoritmos tradicionais de aprendizado de máquina para classificá-los; (ii) técnicas de classificação *end-to-end*. O objetivo final nos dois casos é o mesmo, obter um modelo para predições, resultando em termos de aplicação em um sistema final que faça detecção/diagnóstico/reconhecimento de uma determinada tarefa (ex. câncer, doenças, contaminação de água, etc).

No primeiro caso, as representações (ou vetor de características) computadas pelos métodos desenvolvidos serão analisadas em tarefas de aprendizado supervisionado e não-supervisionado. No aprendizado supervisionado, as representações e rótulos serão usados para treinar algoritmos de classificação bem estabelecidos (por exemplo, *Support Vector Machine* (SVM), *K-Nearest Neighbours* (KNN), etc) e criar modelos para predição de amostras de teste. Neste trabalho, esse método será o considerado, especificamente o LDA (*Linear Discriminante Analysis*).

O (*Linear Discriminante Analysis*) é um método estatístico utilizado para encontrar uma combinação linear de características essenciais para definir uma classe. Sendo assim, esses atributos separaram as classes, sendo usada como classificador linear. Nesse método, as imagens são projetadas em um espaço vetorial, a fim de obter a melhor separação de classes. Dessa forma, as classes terão mais similaridade dentro delas e maior distanciamento entre elas. O LDA utiliza, por exemplo, duas classes C_1 e C_2 , com dados x ; para encontrar a direção definida por um vetor w , de tal forma que x quando projetado em w , separe ao máximo as amostras das duas classes. A projeção de x na direção w é dada pela Eq. 3.2.1 (ALPAYDIN, 2020):

$$z = w^T x. \quad (3.2.1)$$

Assim, a matriz de C_1 é definida pela equação:

$$S_b = (m_1 - m_2)(m_1 - m_2)^T, \quad (3.2.2)$$

enquanto a matriz de C_2 é definida por:

$$S_1 = \sum_{t=1}^N r^t (x^t - m_1)(x^t - m_1)^T. \quad (3.2.3)$$

A matriz S_2 entre as classes é definida da seguinte forma:

$$S_2 = \sum_{t=1}^N (1 - r^t)(x^t - m_2)(x^t - m_2)^T, \quad (3.2.4)$$

em que N é o número total de amostras, x é cada amostra das classes, m_1 e m_2 são as médias das amostras de $C1$ e $C2$ e r^t é um fator de classificação das amostras. A matriz entre as classes total pode ser calculada como o somatório das matrizes das classes, nesse exemplo, sendo a soma das matrizes S_1 e S_2 . Portanto, é possível encontrar a direção do vetor w como demonstrado por:

$$w = S_w^{-1}(m_1 - m_2), \tag{3.2.5}$$

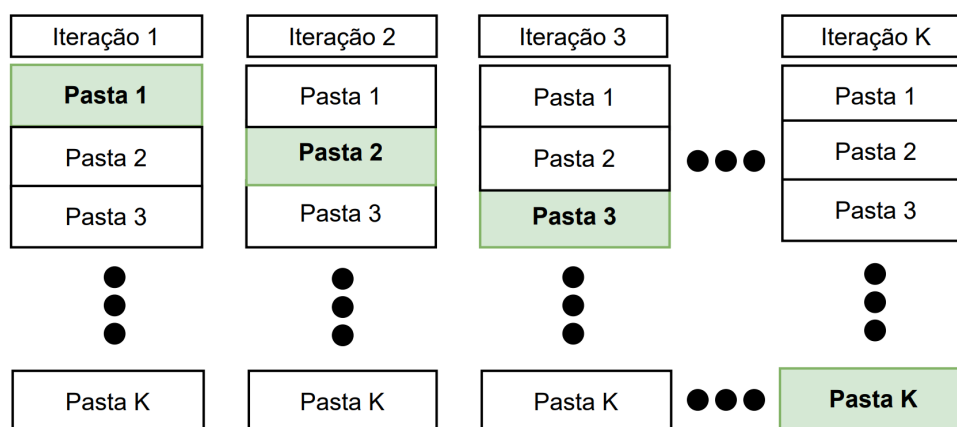
sendo S_w a matriz entre classes total.

3.3 Esquemas de validação

Para avaliar e validar os métodos de análise de imagens e aprendizado de máquina, serão realizados experimentos com *benchmarks* da literatura e conjuntos de dados das aplicações em questão. Para garantir uma avaliação robusta da capacidade de generalização dos modelos, será empregado um procedimento de reamostragem, utilizando a técnica de validação cruzada *leave-one-out*.

Na abordagem *leave-one-out*, cada dado do conjunto é tratado como um subconjunto isolado de validação, enquanto os demais dados são utilizados para o treinamento do modelo. Assim, o treinamento do modelo é repetido tantas vezes quanto o número de amostras no conjunto de dados, sendo que, em cada iteração, uma única amostra é usada para validação, destacada em verde na Figura 3.3.1, e o restante das amostras é usado para o treinamento.

Figura 3.3.1 – Validação cruzada



Fonte: Autoria própria

Esse processo de *leave-one-out* divide o conjunto de imagens em múltiplas iterações, uma para cada amostra, maximizando o uso dos dados para avaliação e treinamento. A performance do modelo é calculada em cada iteração e, ao final, a média dessas avaliações é utilizada como estimativa final da performance do modelo.

Essa estratégia proporciona uma avaliação robusta, pois cada dado é utilizado como amostra de validação exatamente uma vez, e é vantajosa ao fornecer uma estimativa mais confiável da performance do modelo, utilizando todos os dados para treinamento e validação. Além disso, o método ajuda a prevenir o *overfitting*, garantindo que o modelo generalize bem para dados não vistos.

3.4 Métricas de desempenho

Nesta seção serão abordadas as métricas utilizadas para validar os modelos e medir seu desempenho, a fim de garantir a qualidade dos modelos propostos. As métricas estatísticas utilizadas para avaliação de desempenho em tarefas de classificação foram a acurácia, desvio padrão, precisão, sensibilidade e especificidade.

3.4.1 Acurácia

A acurácia de um modelo pode ser calculada como o número total de classificações corretas dividido pelo número total de amostras no conjunto de dados, como mostra a equação a seguir:

$$\text{Acurácia} = \frac{\text{Número de classificações corretas}}{\text{Número total de classificações}} \quad (3.4.1)$$

3.4.2 Desvio padrão

O desvio padrão visa apurar a estabilidade do modelo, uma vez que ele indica a variação entre as acurácias entre as K pastas do *cross-validation*, como demonstra a equação:

$$\sigma = \sqrt{\frac{1}{K-1} \sum_{i=1}^K (\text{acc}_i - \overline{\text{acc}})^2} \quad (3.4.2)$$

em que acc é a acurácia.

3.4.3 Matriz de confusão

Em problemas de classificação, muitas vezes são utilizadas matrizes de confusão para avaliar os modelos. Elas são, essencialmente, uma tabela que indica os erros e acertos do modelo, comparando com o resultado esperado, os rótulos. A Tabela 1 exemplifica uma matriz de confusão.

Sendo os verdadeiros positivos a classificação correta da classe positivo, enquanto os verdadeiros negativos são a classificação correta da classe Negativo. Já as

Tabela 1 – Matriz de confusão

		Detectada	
		Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: Autoria própria

outras duas predições, representadas em verde, são erros. O FN é um erro em que o modelo previu a classe negativo quando o valor real era classe positivo; e o FP é o erro em que o modelo previu a classe positivo quando o valor real era classe negativo. Com base na matriz de confusão podemos calcular as seguintes métricas do modelo:

3.4.3.1 Precisão

A precisão indica quantas classificações estão corretas entre todas as classificações de classe positivo que o modelo fez. Ela pode ser descrita como:

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (3.4.3)$$

3.4.3.2 Sensibilidade

Essa métrica mostra dentre todas as situações de classe positivo como valor esperado, quantas estão corretas. A sensibilidade, ou *recall* é:

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \quad (3.4.4)$$

3.4.3.3 Especificidade

Ela aponta a proporção de verdadeiros negativos, sendo a capacidade do sistema em prever corretamente a classe negativo para casos que realmente são negativos.

$$\text{Especificidade} = \frac{VN}{VN + FP} \quad (3.4.5)$$

Usando essas métricas, as abordagens desenvolvidas e as da literatura serão comparados para avaliar a melhoria e determinar quais são as mais apropriadas para as aplicações. Além disso, os resultados obtidos com as abordagens baseadas em análise de imagens e aprendizado de máquina poderão ser comparados com as demais técnicas usadas, por exemplo, em detecção de osteoartrite existem estudos onde

se utilizam métodos descritores clássicos, e também redes neurais convolucionais com *fine-tuning*, entre outros.

4 Abordagem proposta

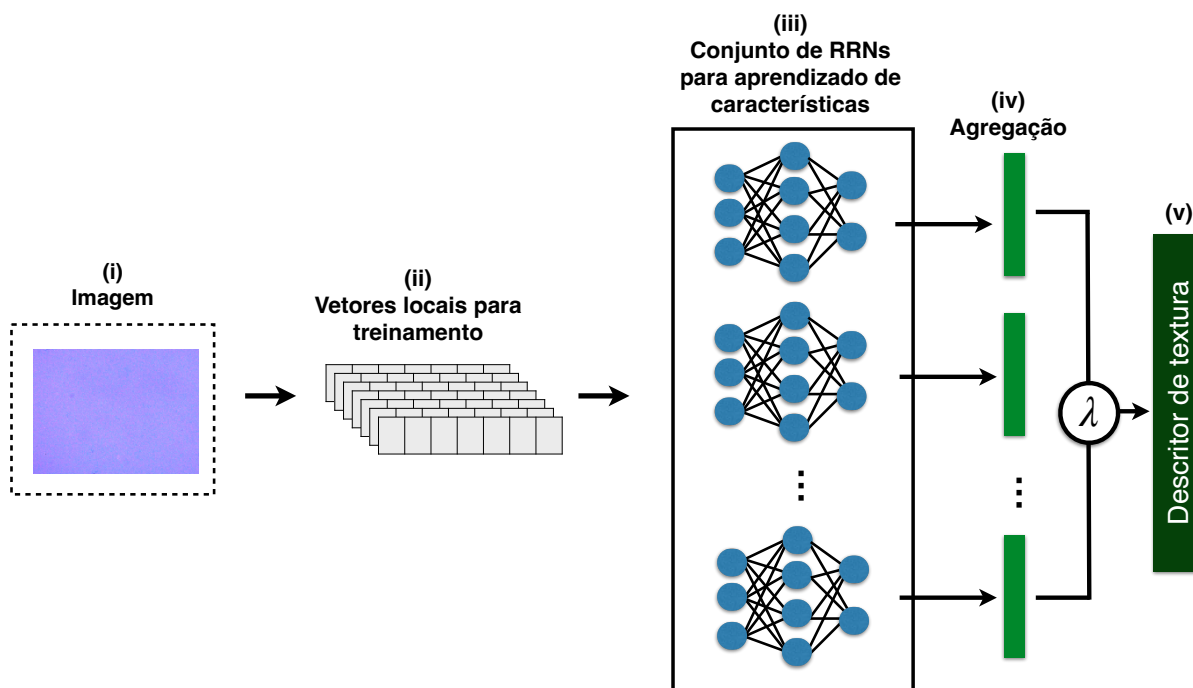
Neste trabalho, propõe-se uma abordagem de caracterização de texturas utilizando uma sopa de modelos de RNNs com aprendizado supervisionado. Baseando-se na estratégia de *ensemble* descrita por Wortsman et al., a metodologia consiste em treinar múltiplos modelos de redes neurais de maneira independente, com inicializações de pesos distintas, e, em seguida, combinar os pesos desses modelos através de métricas como a média, desvio padrão, curtose e assimetria. A ideia central é explorar diferentes projeções do espaço de entrada proporcionadas pelas variações nos pesos aleatórios, o que permite capturar uma gama mais ampla de características das texturas. Esta técnica visa criar uma representação mais robusta e abrangente das texturas, melhorando a acurácia e a capacidade de discriminação em texturas.

No entanto, nossa abordagem apresenta uma modificação significativa em relação ao método original de "sopa de modelos". Neste caso, a acurácia não é recalculada após a adição de cada modelo individual, mas sim após a combinação de modelos com diferentes parâmetros, o que permite avaliar o impacto de combinações específicas de redes. Os parâmetros variáveis incluem a quantidade de neurônios na camada oculta de cada rede neural, as métricas de agregação utilizadas para combinar as saídas das redes (tais como média, desvio padrão, curtose e assimetria), e a quantidade de redes incluídas na sopa.

Esse processo de experimentação com combinações de parâmetros tem o objetivo de identificar configurações que maximizem a capacidade de aprendizado da sopa de modelos, gerando uma estrutura que não apenas capture as características mais discriminativas das texturas, mas que também ofereça maior flexibilidade e eficiência em situações variadas de entrada.

Na etapa de caracterização de texturas, o objetivo é investigar diferentes maneiras de aplicar redes neurais randomizadas em imagens, visando aprimorar a capacidade de aprendizado de características discriminativas para a classificação de texturas. O *framework* para extração de características é ilustrado na Figura 4.4.1, e a seguir são detalhadas as possíveis melhorias e contribuições a serem exploradas em cada um dos passos.

Figura 4.0.1 – Ilustração das etapas da metodologia de aprendizado de características texturas via RNNs



Fonte: Autoria própria

4.1 Entrada das redes

A entrada da rede neural nesta metodologia difere significativamente das abordagens tradicionais de aprendizado de máquina, onde frequentemente se trabalha com conjuntos de dados extensos compostos por múltiplas imagens para treinar modelos. Aqui, a entrada é uma única imagem (i), e o objetivo é extrair uma representação de textura específica dessa imagem em particular.

Esse enfoque singular permite que as RNNs sejam treinadas utilizando os atributos únicos de uma única imagem, o que pode capturar de maneira mais detalhada as características texturais presentes na mesma. A escolha de trabalhar com uma única imagem por vez visa maximizar a capacidade do modelo de entender e representar os detalhes texturais que poderiam ser perdidas em uma abordagem tradicional de conjunto de dados.

4.2 Transformação da imagem em vetores de padrões

Conforme ilustrado na Figura 4.4.1 (ii), a imagem original é transformada em um conjunto de vetores de padrões que servirão como atributos de entrada para as RNNs e um conjunto de rótulos que serão os alvos para o treinamento super-

visionado. Este processo de transformação é essencial para a extração de características relevantes da textura da imagem, permitindo que o modelo aprenda a partir de informações específicas e detalhadas sobre a distribuição espacial dos pixels.

Para realizar essa transformação, a imagem é dividida em pequenas regiões ou janelas, que normalmente possuem dimensões reduzidas, como 3x3 pixels. Cada janela é então processada individualmente para extrair o vetor de atributos e o rótulo correspondente. Por exemplo, ao dividir a imagem em janelas de 3x3 pixels, o nível de cinza do pixel central de cada janela é utilizado como o rótulo de saída, enquanto os níveis de cinza dos 8 pixels vizinhos formam o vetor de entrada para as RNNs. Essa técnica é eficaz porque permite que o modelo capture padrões locais de textura, como bordas, cantos e outras características distintivas.

Além das janelas de 3x3 pixels, também são testadas janelas de 5x5 e 7x7 pixels, que oferecem uma perspectiva mais ampla sobre os padrões texturais da imagem. As janelas de 5x5 e 7x7 permitem incluir mais contexto ao redor de cada pixel central, capturando variações mais sutis e complexas nas texturas. Isso é particularmente útil para identificar padrões texturais maiores ou mais difusos, que podem não ser adequadamente representados por janelas menores.

Além de testar separadamente as janelas de 3x3, 5x5 e 7x7 pixels, exploramos também a combinação dos vetores obtidos a partir dessas configurações. Ao combinar os vetores de 3x3, 5x5 e 7x7, é possível criar uma representação ainda mais rica e diversa da textura, beneficiando-se das vantagens de ambas as janelas. Enquanto a janela de 3x3 foca em detalhes finos e específicos, as janelas de 5x5 e 7x7 proporcionam um entendimento mais amplo e contextual.

4.3 Treinamento das RNNs

Após a obtenção dos vetores de padrões locais (atributos) e seus respectivos rótulos, o próximo passo é empregar as RNNs para o aprendizado das representações texturais (iii). Para maximizar a captura das diferentes propriedades texturais, utilizamos um *ensemble* de RNNs, empregando a estratégia de sopa de modelos descrita por Wortsman et al. (WORTSMAN et al., 2022). Cada RNN é inicializada com diferentes pesos aleatórios e treinada independentemente, explorando diferentes regiões do espaço de hipóteses. Essa diversidade no treinamento permite que cada RNN capture diferentes aspectos das texturas presentes nas imagens.

Após o treinamento das RNNs, os resultados são combinados de maneira a criar um modelo final robusto e abrangente. Essa combinação é realizada através de diversas operações estatísticas sobre os pesos das RNNs treinadas. Essa abordagem de agrupamento diversificado permite que o modelo final incorpore os melhores

aspectos de cada RNN individual, resultando em uma representação textural mais rica e precisa.

4.4 Representações

Ao final do treinamento, cada RNN fornece como saída uma representação que define as *features* extraídas. Finalmente, para obter a representação final com os descritores de textura, é essencial agregar as diferentes representações geradas pelas RNNs em um único vetor que capture as características texturais de maneira abrangente (iv). Essa representação final é construída com base nos pesos aprendidos pela camada de saída das RNNs, que representam as características extraídas dos padrões locais da imagem.

Assim, a camada de entrada de cada RNN desse conjunto é inicializada com pesos aleatórios diferentes. Para isso, a sequência aleatória V da Equação 2.1.3 é gerada com comprimento $E * z$, em que z é o número de redes usadas na sopa de modelos e $E = Q * (p + 1)$ como definido na Seção 2.1.1. Em seguida, a sequência V é dividida em z partes, sendo cada parte usada para construir uma matriz de pesos aleatórios W_i da camada de entrada de uma RNN r_i da sopa:

$$W_1 = \{V(1), \dots, V(E)\} \quad (4.4.1)$$

$$W_2 = \{V(E + 1), \dots, V(2 * E)\} \quad (4.4.2)$$

$$W_z = \{V(((z - 1) * E) + 1), \dots, V(z * E)\}. \quad (4.4.3)$$

Afim de entender as diferentes representações que a sopa de modelos pode gerar para cada imagem, foi realizado um estudo em que diferentes medidas foram calculadas para agregar o conjunto de pesos treinados M . As medidas estudadas foram a média aritmética, o desvio padrão, a curtose e a assimetria.

Seja $\vec{x} = \{x_1, x_2, \dots, x_N\}$ um conjunto de N números reais, então a média é dada por:

$$\mu(\vec{x}) = \frac{1}{N} \sum_{k=1}^N x_k \quad (4.4.4)$$

O desvio padrão é dado por:

$$\sigma(\vec{x}) = \sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_k - \mu(\vec{x}))^2} \quad (4.4.5)$$

A assimetria é dada por:

$$\gamma(\vec{x}) = \frac{\frac{1}{N} \sum_{k=1}^N (x_k - \mu(\vec{x}))^3}{\left(\sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_k - \mu(\vec{x}))^2} \right)^3} \quad (4.4.6)$$

A curtose é dada por:

$$\kappa(\vec{x}) = \frac{\frac{1}{N} \sum_{k=1}^N (x_k - \mu(\vec{x}))^4}{\left(\sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_k - \mu(\vec{x}))^2} \right)^4} \quad (4.4.7)$$

Assim, essas medidas foram calculadas dos vetores $\{M_1, \dots, M_z\}$ da seguinte forma para compor a representação final de textura:

$$\beta_\sigma = \sigma \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_z \end{pmatrix}, \quad \beta_\mu = \mu \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_z \end{pmatrix} \quad (4.4.8)$$

$$\beta_\gamma = \gamma \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_z \end{pmatrix}, \quad \beta_\kappa = \kappa \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_z \end{pmatrix} \quad (4.4.9)$$

em que β_σ , β_μ , β_γ e β_κ são os vetores de características obtido pelas medidas desvio, média, a assimetria e a curtose, respectivamente. Como as RNNs são treinadas usando dois tamanhos de janelas diferentes, o vetor V_Q^ϕ é definido como a combinação de uma determinada medida calculada da sopa de RNNs treinadas usando os tamanhos de janela 3×3 e 5×5 :

$$V_Q^\phi = [\beta_\phi^3, \beta_\phi^5], \quad (4.4.10)$$

onde ϕ representa uma das medidas da sopa (média, desvio, assimetria, curtose) e Q o número de neurônios das camadas escondida das RNNs.

Ademais, a sopa pode ser realizada com RNNs com diferentes quantidades de neurônios ocultos - Q . Nesse estudo, Q usou-se os valores $Q = \{9, 19, 39, 59, 69, 89, 99\}$ e suas combinações em duplas e trios:

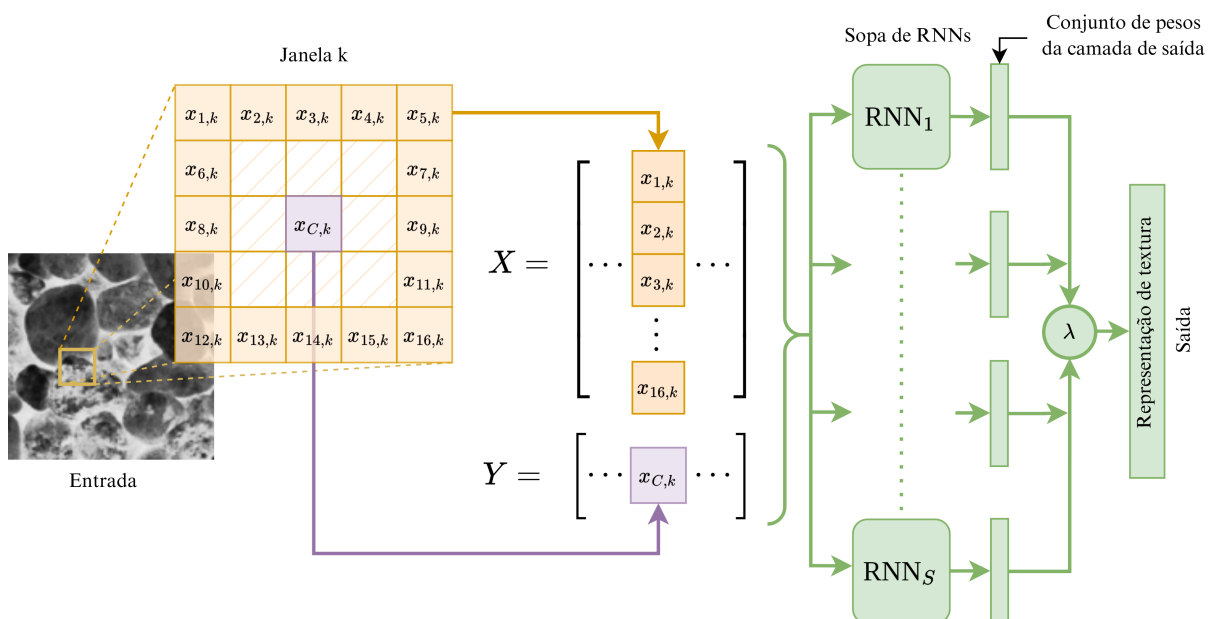
$$\varphi_{\{Q_1, \dots, Q_m\}}^\phi = [V_{Q_1}^\phi, \dots, V_{Q_m}^\phi], \quad (4.4.11)$$

Por fim, a combinação entre os vetores $\varphi_{\{Q_1, \dots, Q_m\}}^\phi$ para diferentes métricas também foi investigada para compor a representação final.

A média dos pesos é usada para suavizar as características individuais das RNNs, criando uma representação geral mais estável das texturas, enquanto o desvio padrão oferece uma medida da variabilidade entre as representações aprendidas, ajudando a compreender a dispersão dos dados. A curtose e a assimetria foram consideradas para capturar a distribuição dos pesos, fornecendo *insights* sobre a presença de *outliers* e a inclinação dos dados. Essas estratégias de agregação foram selecionadas para maximizar a discriminação entre diferentes texturas, assegurando que a representação final capture uma gama abrangente de características texturais.

A Figura 4.4.1 demonstra o aprendizado de representações utilizando RNNs para compor a sopa de modelos. A entrada do método é uma imagem de textura que será processada. Assim, uma janela deslizante k é extraída da imagem de entrada, onde cada janela contém vários pixels $x_{i,k}$. Em seguida cada janela é vetorizada, transformando a matriz de pixels em um vetor X contendo os pixels $x_{1,k}, x_{2,k}, \dots$. O pixel central $x_{C,k}$ da janela é isolado para formar o vetor Y . A sopa ocorre quando uma coleção de RNNs é aplicada ao vetor X . Cada RNN gera uma representação de textura a partir do vetor de entrada. As saídas das diferentes RNNs são combinadas através de uma função de agregação λ , resultando em uma representação de textura final.

Figura 4.4.1 – Sopa de modelos para aprendizado de representações de textura



Fonte: Adaptado de (FARES et al., 2024).

5 Resultados

A seguir são apresentados os resultados experimentais obtidos a partir da aplicação da abordagem de caracterização de texturas utilizando uma sopa de RNNs. Esta seção detalha o desempenho do modelo proposto em termos de capacidade de discriminação de diferentes texturas, avaliando a eficácia das diferentes estratégias de agregação aplicadas, como a média, desvio padrão, curtose e assimetria dos pesos das RNNs. Adicionalmente, são comparados os resultados para diferentes configurações de janelas de extração de padrões, a quantidade Q de neurônios ocultos e o número de redes neurais na sopa, assim como suas combinações.

5.1 Avaliação de parâmetros

Para avaliar o método proposto, foram testadas diferentes combinações de parâmetros visando maximizar a precisão e a representatividade da sopa de modelos. Os parâmetros considerados incluem a quantidade de neurônios Q na camada oculta das RNNs, o número de redes m (5, 9, 10, 15, 18, 20, 27, 36, 45, 54) utilizadas na sopa, e as métricas de agregação (média, desvio padrão, assimetria e curtose) aplicadas aos pesos dos modelos, que servem para compor as representações finais.

A análise foi conduzida usando raios de diferentes tamanhos (3x3, 5x5 e 7x7), considerando também suas combinações, de modo a captar tanto características micro quanto macro das amostras de textura. Para explorar a melhor configuração dos neurônios na camada oculta, foram testados valores de Q nos conjuntos {9, 19, 29, 39, 49, 59}, avaliados individualmente e em pares, para examinar se configurações combinadas de neurônios poderiam capturar padrões mais complexos nos dados.

Na Tabela 2, foram calculadas as médias de acurácia para uma sopa de 10 RNNs ($m = 10$), testada nos conjuntos de dados Outex, Brodatz e Usptex. Essa análise permitiu observar a influência das combinações de Q e das métricas de agregação sobre o desempenho. Constatou-se que pares de valores para Q , como [9, 29], proporcionaram melhores resultados, destacando-se por capturar características complementares das texturas.

Em relação às métricas de agregação, combinações entre todas elas também demonstraram maior eficácia. As representações geradas a partir da combinação de todas as métricas de agregação (β_σ , β_μ , β_γ e β_κ) apresentaram resultados superiores, alcançando uma acurácia média de 95,54%. Esses resultados indicam que a combinação dos parâmetros aumenta significativamente a robustez e a capacidade

discriminativa da sopa de modelos, sugerindo que o uso de pares de neurônios na camada oculta e múltiplas métricas de agregação é vantajoso para a classificação de texturas.

Tabela 2 – Média das acurácias das bases Outex, Brodatz e Usptex, ao variar as medidas e Q , utilizando 5 redes neurais, e raio [3, 5, 7]

Medidas	Q	9	19	29	39	49	59
	Média		89,24	91,31	92,03	92,05	92,38
Desvio		85,90	89,12	89,69	90,31	90,43	86,02
Curtose		80,64	85,61	86,47	86,82	87,00	84,26
Assimetria		81,63	86,57	87,90	88,99	89,36	87,32
Média + Assimetria		90,88	92,23	93,42	93,26	93,35	91,87
Média + Curtose		90,97	92,78	92,85	93,05	93,24	91,62
Média + Desvio		91,21	92,89	93,62	93,09	92,92	90,04
Média + Desvio + Assimetria		92,35	93,15	93,75	93,88	93,79	92,05
Média + Desvio + Curtose		92,04	93,49	94,04	93,54	93,50	91,73
Média + Desvio + Assimetria + Curtose		92,56	93,54	94,40	93,88	93,55	91,91

Tabela 3 – Acurácias das bases Outex, Brodatz e Usptex, e a média entre elas, ao variar Q e suas combinações até dois valores, fixando raio [3, 5, 7] e $m = 10$. A linha enfatizada indica a representação da textura com maior precisão média.

$\{Q_1, Q_2, \dots, Q_m\}$	Nº Features	Brodatz	USPtex	Outex	Média
{09}	120	95,21	93,89	90,51	93,20
{19}	240	95,72	94,85	91,76	94,11
{29}	360	95,89	96,38	91,10	94,46
{39}	480	96,62	94,85	90,15	93,87
{49}	600	96,11	94,59	88,24	92,98
{59}	720	96,17	95,03	88,38	93,19
{09, 19}	360	96,34	95,72	91,03	94,36
{09, 29}	480	97,07	96,99	92,57	95,54
{09, 39}	600	96,96	95,99	90,37	94,44
{09, 49}	720	96,28	96,07	90,44	94,26
{09, 59}	840	96,68	96,12	89,71	94,17
{19, 29}	600	96,57	96,86	91,25	94,89
{19, 39}	720	97,02	96,38	90,51	94,64

O último parâmetro experimentado, o número de redes, atingiu as melhores porcentagens com 10 redes neurais na sopa, como demonstra a Tabela 4. Apesar de aumentar a quantidade de redes na sopa, os resultados não melhoram, demonstrando que há uma saturação dos resultados, tendo em vista que mais redes neurais não apresentaram maiores acurácias.

5.2 Comparações com a literatura

Com o objetivo de apurar o desempenho da abordagem desenvolvida, neste estudo são comparadas as acurácias atingidas ao utilizar janelas de pixels de tamanho

Tabela 4 – Acurácias das bases Outex, Brodatz e Usptex, ao variar a quantidade de redes neurais m , utilizando $Q = \{9, 29\}$ e raio $[3, 5, 7]$

m	Brodatz	USPtex	Outex	Média
5	96,73	95,55	92,43	94,90
10	97,07	96,99	92,50	95,52
15	96,90	96,03	91,03	94,65
18	95,50	94,20	86,91	92,20
45	96,28	95,86	89,63	93,92

$\{3, 5, 7\}$ combinados, quantidade de neurônios $\{9, 29\}$, conjunto de medidas $\{\text{média, desvio, curtose e assimetria}\}$ e $m = 10$ RNNs na sopa, com outros métodos de análise de texturas da literatura. Essa configuração de parâmetros foi a que obteve o melhor equilíbrio entre acurácia e tamanho do vetor de características nas três base de dados estudadas. Na Tabela 5 o número de *features* foi levado em consideração também. Na perspectiva das três bases apuradas, a sopa de modelos RNN proposta - que continha 10 redes neurais - obteve um bom desempenho, tendo em vista que seus resultados nas três colunas foram os superiores em comparação aos demais métodos. Ademais, nota-se que mesmo os métodos que utilizaram assinaturas maiores, como os métodos LBPV e CLBP resultaram em acurácias menores do que a sopa de RNNs, que produz 480 *features* na combinação fixada.

Outro ponto importante é a comparação com redes neurais profundas, que utilizam muito mais *features* como entrada, comparado com a sopa. Além disso, essas redes comumente necessitam de um processamento extensivo, exigindo máquinas robustas e tempo de execução lento. A sopa de RNNs, por sua vez, pode ser utilizada em um computador pessoal e treinada rapidamente.

De forma análoga, a Tabela 6 apresenta os resultados da base 1200tex, que é uma base de reconhecimento de espécies de plantas baseado em textura foliar. Novamente a porcentagem da acurácia adquirida pela sopa de RNNs é superior aos outros métodos. É essencial notar que todos os métodos comparados utilizaram o LDA como técnica para classificação dos *features*, além de utilizar as imagens em escala de cinza, tal qual o método utilizado nesse estudo.

Também é notório que quando comparado ao ELM *signature* (JUNIOR; BACKES, 2016), cuja ideia é utilizar uma única RNN, a proposta de sopa de modelos também obteve maiores acurácias nas três bases de dados. Além disso, a sopa de modelos superou os resultados de todos os outros métodos, registrando ganhos de mais de 30%, como observado na base USPTex quando comparado ao método Fourier. Essa tendência reforça a eficácia da estratégia proposta de combinar múltiplas redes neurais para análise de texturas.

Tabela 5 – Comparação de acurácias de diferentes métodos de análise de textura em três bancos de dados de texturas.

Métodos	Nº de features	Outex	USPTex	Brodatz
GLDM (WESZKA; DYER; ROSENFELD, 1976)	60	86,76	92,06	94,43
Gabor Filters (MANJUNATH; MA, 1996)	48	81,91	89,22	89,86
Fourier (WESZKA; DYER; ROSENFELD, 1976)	63	81,91	67,50	75,90
Fractal (BACKES; CASANOVA; BRUNO, 2009)	69	80,51	78,27	87,16
LBP (OJALA; PIETIKAINEN; MAENPAA, 2002)	256	81,10	85,43	93,64
CLBP (GUO; ZHANG; ZHANG, 2010)	648	85,80	91,14	95,32
AHP (ZHU et al., 2015)	120	88,31	94,85	94,88
LCP (GUO; ZHAO; PIETIKÄINEN, 2011)	81	86,25	91,14	93,47
LFD (MAANI; KALRA; YANG, 2013)	276	82,57	83,55	90,99
LPQ (OJANSIVU; HEIKKILÄ, 2008)	256	79,41	85,12	92,51
ELM Signature (JUNIOR; BACKES, 2016)	180	89,71	95,11	95,27
CNTD (BACKES; CASANOVA; BRUNO, 2013)	108	86,76	91,71	95,27
CNRNN (RIBAS et al., 2020)	240	91,32	96,95	96,06
SSR ¹ (RIBAS et al., 2024a) (gray-scale)	630	90,80	95,80	–
VGG19 (SIMONYAN; ZISSERMAN, 2014)	512	76,62	93,19	96,79
InceptionV3 (SZEGEDY et al., 2016)	2048	86,40	96,77	98,54
ResNet50 (HE et al., 2016)	2048	65,66	62,30	81,98
InceptionResNetV2 (SZEGEDY et al., 2017)	1536	85,88	96,34	98,99
Abordagem proposta				
$\bar{\Omega}(09, 29)$	480	92,57	96,99	97,07

Tabela 6 – Acurácia usando o classificador LDA na base de dados 1200Tex.

Métodos	Acurácias (%)
AHP (ZHU et al., 2015)	79,17
GLDM (WESZKA; DYER; ROSENFELD, 1976)	79,92
LCP (GUO; ZHAO; PIETIKÄINEN, 2011)	76,58
LFD (MAANI; KALRA; YANG, 2013)	74,67
LPQ (OJANSIVU; HEIKKILÄ, 2008)	73,00
Fourier (WESZKA; DYER; ROSENFELD, 1976)	65,75
Fractal (BACKES; CASANOVA; BRUNO, 2009)	70,75
Gabor (MANJUNATH; MA, 1996; DAUGMAN; DOWNING, 1995)	77,25
CNTD (BACKES; CASANOVA; BRUNO, 2013)	83,33
ELM signature (JUNIOR; BACKES, 2016)	84,86
Sopa de RNN (proposta neste estudo)	87,42

5.3 Diagnóstico médico de osteoartrite

A osteoartrite (OA) é uma condição das articulações caracterizada pela degeneração da cartilagem e mudanças nos ossos. Identificar precocemente a OA do joelho por meio da análise da textura óssea em imagens médicas é um desafio, tendo em vista a dificuldade visual para detectar a doença, como pode ser observado na Figura 5.3.1. Nesse estudo, foram utilizadas imagens de raio-X do joelho provenientes do banco de dados da OsteoArthritis Initiative (OAI) para realizar uma classificação entre joelhos com OA e sem OA, com o objetivo de avaliar a precisão do modelo desenvolvido nesse projeto. A base de dados utilizada conta com 688 imagens de joelho, sendo

metade destes indivíduos saudáveis, e metade com osteoartrite. O experimento contou com duas classes, considerando o joelho normal (K&L grau 0) e OA mínima (K&L grau 2).

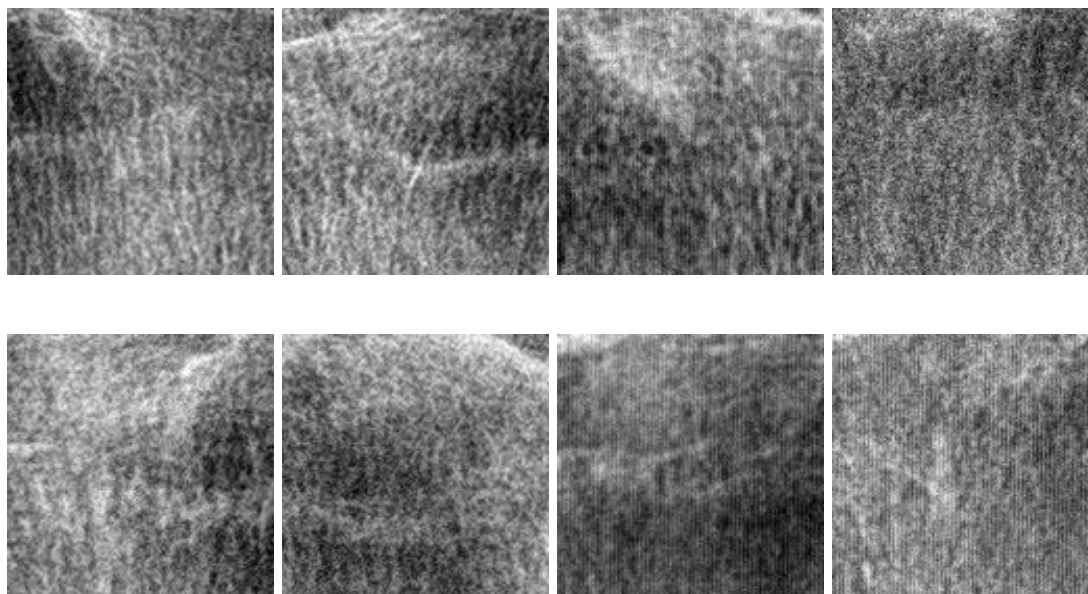


Figura 5.3.1 – Exemplos de imagens de joelho sem OA (primeira linha) e com OA mínima (segunda linha).

A Tabela 7 conta com métodos descritores clássicos, que buscam extrair características específicas da imagem para representar suas propriedades. Também foram usadas redes neurais convolucionais com *fine-tuning*, onde se ajusta uma rede pré-treinada para aprender características gerais, e em seguida, modifica-se os pesos da rede em um conjunto de dados menor e mais específico. Além disso, ao utilizar as CNNs, testou-se também reduzir os parâmetros das camadas convolucionais através de uma estratégia GAP, e logo após foi utilizado um classificador SVM. Dessa forma, ao utilizar os parâmetros que obtiveram melhor desempenho nessa base, sendo $Q = 9$, raio [3, 5, 7], a agregação de todas as medidas em uma sopa de 10 RNNS, o resultado obtido foi de 79,36% superando os resultados obtidos por trabalhos anteriores.

Tabela 7 – 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	Sopa de RNNs	81,69	77,03	79,36

6 Conclusão

Neste trabalho de conclusão de curso, foi possível aprofundar o estudo em métodos de análise de textura e o uso de Redes Neurais Randomizadas (RNNs) nessa tarefa. Inicialmente, foi realizada uma investigação detalhada sobre o processo de caracterização de imagens texturizadas, abordando as técnicas de extração de características que melhor representam as propriedades de textura em imagens digitais. Em seguida, foi aprofundado o conhecimento sobre aprendizado e treinamento de RNNs, com foco em sua aplicação na classificação de texturas. Além disso, foram estudadas diversas técnicas de classificação e validação, bem como métodos avançados, como as técnicas de *ensemble*, para melhorar a robustez e a precisão dos modelos.

A principal proposta desenvolvida neste trabalho para análise de texturas foi a aplicação de uma estratégia de sopa de modelos, que visa melhorar as representações geradas pelas RNNs. Essa abordagem consiste em combinar os pesos treinados de várias RNNs, que foram inicializadas com diferentes pesos aleatórios na camada de entrada. As RNNs geram representações variadas das características de textura, e a sopa de modelos explora essa diversidade para obter um modelo mais robusto e preciso. Foram investigadas diversas medidas de agregação para os pesos, incluindo a média, o desvio padrão e a soma, com o objetivo de encontrar a combinação que proporciona os melhores resultados.

Os experimentos realizados demonstraram que a estratégia de sopa de modelos aumentou significativamente a acurácia da classificação em comparação com o uso de uma única RNN. Além disso, essa abordagem superou outros métodos descritos na literatura, que muitas vezes são mais custosos computacionalmente. Este trabalho também contribuiu de maneira prática para o campo de diagnóstico médico. O método proposto foi aplicado em uma base de imagens para detecção de osteoartrite precoce no joelho e mostrou-se eficaz ao aumentar a acurácia em mais de 10% em comparação com outros métodos da literatura. Isso é especialmente relevante porque o aumento da acurácia foi alcançado com um custo computacional relativamente baixo, demonstrando a eficiência e viabilidade da abordagem de sopa de RNNs para aplicações médicas.

Portanto, este trabalho de conclusão de curso não apenas avançou as metodologias computacionais para análise de texturas em imagens, mas também teve um impacto significativo nas aplicações práticas relacionadas ao diagnóstico de doenças. Essa pesquisa também teve um papel fundamental na formação acadêmica da autora, preparando-a para prosseguir com estudos avançados na pós-graduação.

Em advento desse trabalho, foi publicado um artigo na conferência internacional IWSSIP 2024 (FARES et al., 2024). Adicionalmente, outro artigo está em fase de redação para submissão futura, o que reforça a relevância deste trabalho no campo da análise de texturas.

Referências

- ALPAYDIN, E. *Introduction to machine learning*. [S.l.]: MIT press, 2020.
- ATTALLAH, O. Gabrop: gabor wavelets-based cad for retinopathy of prematurity diagnosis via convolutional neural networks. *Diagnostics*, MDPI, v. 13, n. 2, p. 171, 2023.
- BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. 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.
- BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Color texture analysis based on fractal descriptors. *Pattern Recognition*, Elsevier, v. 45, n. 5, p. 1984–1992, 2012.
- BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Texture analysis and classification: A complex network-based approach. *Information Sciences*, Elsevier, v. 219, p. 168–180, 2013.
- BAHROUN, S.; ABED, R.; ZAGROUBA, E. Deep 3d-lbp: Cnn-based fusion of shape modeling and texture descriptors for accurate face recognition. *The Visual Computer*, Springer, v. 39, n. 1, p. 239–254, 2023.
- BRODATZ, P. et al. *Textures: a photographic album for artists and designers*. [S.l.]: Dover publications New York, 1966. v. 2.
- 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 Online Library, v. 19, n. 3, p. 236–243, 2009.
- 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.
- DIETTERICH, T. G. Ensemble methods in machine learning. In: SPRINGER. *International workshop on multiple classifier systems*. [S.l.], 2000. p. 1–15.
- FADAEI, S.; HOSSEINI, P.; RAHIMIZADEH, K. New texture descriptor based on improved orthogonal difference local binary pattern. In: IEEE. *2023 6th international conference on pattern recognition and image analysis (IPRIA)*. [S.l.], 2023. p. 1–5.
- FARES, R. T.; RIBAS, L. C. A new approach to learn spatio-spectral texture representation with randomized networks: Application to brazilian plant species identification. In: SPRINGER. *International Conference on Engineering Applications of Neural Networks*. [S.l.], 2024. p. 435–449.
- FARES, R. T. et al. Randomized encoding ensemble: A new approach for texture representation. In: IEEE. *2024 31st International Conference on Systems, Signals and Image Processing (IWSSIP)*. [S.l.], 2024. p. 1–8.

- FLORINDO, J. B. Fractal pooling: A new strategy for texture recognition using convolutional neural networks. *Expert Systems with Applications*, Elsevier, v. 243, p. 122978, 2024.
- GHAHFARROKHI, S. S. et al. Malignant melanoma diagnosis applying a machine learning method based on the combination of nonlinear and texture features. *Biomedical Signal Processing and Control*, Elsevier, v. 80, p. 104300, 2023.
- GONG, L. et al. Fractal characteristics of the spatial texture in traditional miao villages in qiandongnan, guizhou, china. *Sustainability*, MDPI, v. 15, n. 17, p. 13218, 2023.
- GUO, Y.; ZHAO, G.; PIETIKÄINEN, M. Texture classification using a linear configuration model based descriptor. In: CITESEER. *BMVC*. [S.l.], 2011. p. 1–10.
- 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.
- HARALICK, R.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472.
- HAYKIN, S. *Neural networks and learning machines*, 3/E. [S.l.]: Pearson Education India, 2009.
- HE, K. et al. 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.
- HOSNY, K. M. et al. Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern. *IEEE Access*, IEEE, v. 11, p. 62307–62317, 2023.
- JUNIOR, J. J. d. M. S.; BACKES, A. R. Elm based signature for texture classification. *Pattern Recognition*, Elsevier, v. 51, p. 395–401, 2016.
- 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.
- JUNIOR, J. J. d. M. S.; CORTEZ, P. C.; BACKES, A. R. Color texture classification using shortest paths in graphs. *IEEE Transactions on Image Processing*, IEEE, v. 23, n. 9, p. 3751–3761, 2014.
- JUNIOR, J. J. d. M. S.; 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.
- JUNIOR, J. J. de M. S.; BACKES, A. R.; BRUNO, O. M. Pap-smear image classification using randomized neural network based signature. In: MENDOZA, M.; VELASTÍN, S. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Cham: Springer International Publishing, 2018. p. 677–684.
- JUNIOR, J. J. M. S.; BACKES, A. R. ELM based signature for texture classification. *Pattern Recognition*, v. 51, p. 395–401, 2016.

KAUR, R.; SINGH, S. A comprehensive review of object detection with deep learning. *Digital Signal Processing*, Elsevier, v. 132, p. 103812, 2023.

LIU, L. et al. From bow to cnn: Two decades of texture representation for texture classification. *International Journal of Computer Vision*, Springer, v. 127, n. 1, p. 74–109, 2019.

LUO, Y. et al. Texture classification combining improved local binary pattern and threshold segmentation. *Multimedia Tools and Applications*, Springer, v. 82, n. 17, p. 25899–25916, 2023.

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.

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.

MELEKOODAPPATTU, J. G. et al. Breast cancer detection in mammogram: Combining modified cnn and texture feature based approach. *Journal of Ambient Intelligence and Humanized Computing*, Springer, v. 14, n. 9, p. 11397–11406, 2023.

NANNI, L.; GHIDONI, S.; BRAHNAM, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern recognition*, Elsevier, v. 71, p. 158–172, 2017.

OAI Online. 2018–11–14. <<https://data-archive.nimh.nih.gov/oai/>>.

OJALA, T. et al. Outex-new framework for empirical evaluation of texture analysis algorithms. In: IEEE. *2002 International Conference on Pattern Recognition*. [S.l.], 2002. v. 1, p. 701–706.

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.

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.

PARADOWSKA-STOLARZ, A. et al. Mechanical properties, fractal dimension, and texture analysis of selected 3d-printed resins used in dentistry that underwent the compression test. *Polymers*, MDPI, v. 15, n. 7, p. 1772, 2023.

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.

PATEL, R. K.; KASHYAP, M. Automated diagnosis of covid stages using texture-based gabor features in variational mode decomposition from ct images. *International Journal of Imaging Systems and Technology*, Wiley Online Library, v. 33, n. 3, p. 807–821, 2023.

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.

- RIBAS, L. C. et al. Fusion of complex networks and randomized neural networks for texture analysis. *Pattern Recognition*, v. 103, p. 107189, 2020. ISSN 0031-3203.
- RIBAS, L. C. et al. Color-texture classification based on spatio-spectral complex network representations. *Physica A: Statistical Mechanics and its Applications*, p. 129518, 2024. ISSN 0378-4371.
- RIBAS, L. C. et al. Local complex features learned by randomized neural networks for texture analysis. *Pattern Analysis and Applications*, Springer, v. 27, n. 1, p. 23, 2024.
- RIBAS, L. C. et al. Learning graph representation with randomized neural network for dynamic texture classification. *Applied Soft Computing*, v. 114, p. 108035, 2022. ISSN 1568-4946.
- RODRIGUES, V. C. 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. ISSN 0039-9140. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0039914020307359>>.
- SCABINI, L. et al. A comparative survey of vision transformers for feature extraction in texture analysis. *arXiv preprint arXiv:2406.06136*, 2024.
- SILVA, J. et al. Structural mri texture analysis for detecting alzheimer's disease. *Journal of Medical and Biological Engineering*, Springer, v. 43, n. 3, p. 227–238, 2023.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <<http://arxiv.org/abs/1409.1556>>.
- SOARES, J. C. et al. Detection of a SARS-CoV-2 Sequence with Genosensors Using Data Analysis Based on Information Visualization and Machine Learning Techniques. 12 2020. Disponível em: <https://chemrxiv.org/articles/preprint/Detection_of_a_SARS-CoV-2_Sequence_with_Genosensors_Using_Data_Analysis_Based_on_Information_Visualization_and_Machine_Learning_Techniques/13366379>.
- SZEGEDY, C. et al. 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.
- SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826.
- TUSSUPOV, J. et al. Applying machine learning to improve a texture type image. *Eastern-European Journal of Enterprise Technologies*, v. 122, n. 2, 2023.
- 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.
- WORTSMAN, M. et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2022. p. 23965–23998.

ZHOU, Z.-H.; ZHOU, Z.-H. *Ensemble learning*. [S.l.]: Springer, 2021.

ZHU, Z. et al. An adaptive hybrid pattern for noise-robust texture analysis. *Pattern Recognition*, Elsevier, v. 48, n. 8, p. 2592–2608, 2015.