



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA

SOLON ALVES PEIXOTO

TRANSFORMADA *FLOOR OF LOG* APLICADA EM CONTEXTO
CROSS-DIMENSIONAL

FORTALEZA

2023

SOLON ALVES PEIXOTO

TRANSFORMADA *FLOOR OF LOG* APLICADA EM CONTEXTO CROSS-DIMENSIONAL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Pedro Pedrosa Rebouças Filho

FORTALEZA

2023

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

P431t Peixoto, Solon Alves.

Transformada floor of log aplicada em contexto cross-dimensional / Solon Alves Peixoto. – 2023.
84 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2023.

Orientação: Prof. Dr. Pedro Pedrosa Rebouças Filho.

1. Compressão de características. 2. Algoritmo de agrupamento supervisionado. 3. Reconhecimento facial. 4. Segmentação de pulmão. 5. Aprendizado profundo. I. Título.

CDD 621.38

SOLON ALVES PEIXOTO

TRANSFORMADA *FLOOR OF LOG* APLICADA EM CONTEXTO CROSS-DIMENSIONAL

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 05 de Julho de 2023

BANCA EXAMINADORA

Prof. Dr. Pedro Pedrosa Rebouças
Filho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará (UFC)

Prof. Dr. Paulo Antônio Leal Rego
Universidade Federal do Ceará (UFC)

Prof. Dr. Rodrigo de Melo Souza Veras
Universidade Federal do Piauí (UFPI)

Prof. Dr. Mário Wedney de Lima Moreira
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará (IFCE)

Prof. Dr. Glendo de Freitas Guimarães
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará (IFCE)

À minha família.

AGRADECIMENTOS

Para a minha querida família Geísa e Gustavo, cuja paciência e apoio incondicional foram meu sustento nos tempos mais difíceis desta jornada;

Aos meus pais Márcia Cristina e Alexandre Domingues, pelo amor e sacrifício constante além do incentivo e ensinamentos para a vida;

Ao meu orientador, Prof. Dr. Pedro Pedrosa Rebouças Filho, minha sincera gratidão por iluminar o caminho da minha trajetória acadêmica com sabedoria, serenidade e seriedade;

Ao Laboratório de Processamento de imagens, sinais e computação aplicada (LAPISCO) e todos os seus membros, pelas calorosas discussões científico-filosóficas durante o curso;

Fácil, depois de feito.

(Provérbio popular)

RESUMO

Este trabalho propõe um método de agrupamento, independente de dimensionalidade, chamado transformada *Floor of Log* (FoL). A vantagem deste método consiste na sua facilidade e praticidade na implementação, como também na capacidade de poder gerar múltiplos efeitos nos dados, entre eles, destacam-se a clusterização e a compressão. Foram escolhidas três aplicações direcionadas para soluções de problemas reais. Na primeira foi avaliado o FoL em tarefas relacionadas a reconhecimento facial, especificamente sobre os arrays das *features*. Para essa avaliação, foram utilizados os datasets CelebA, Extended YaleB, AR e LFW em conjunto com a análise do tamanho do dataset após a aplicação do FoL e da acurácia (ACC) do resultado de *matching* entre as faces. A segunda aplicação avalia o FoL em um ambiente bi-dimensional, especificamente em tomografia computadorizada para segmentação de pulmão e, conseqüentemente, dentro de um ambiente de processamento de imagens. Nesta avaliação, foram utilizados os datasets LUNA16 e LAPISCO em conjunto das métricas Hausdorff Distance (HD), DICE, ACC, Jaccard e *Matthews Correlation Coefficient* (MCC). A terceira aplicação busca avaliar o FoL em um contexto mais independente de dimensão, dentro de redes neurais convolucionais para propósitos gerais. Foram utilizados os datasets de *benchmark* CIFAR10 e CIFAR100 além de Davies-Bouldin (DB), Calinski-Harabasz (CH) e Silhouette (Sil). Como resultados, o FoL quando aplicado sobre os arrays nos datasets CelebA, Extended YaleB, AR e LFW, obteve resultados iguais ou melhores quando comparados com a abordagem usando os mesmos classificadores com características não comprimidas, mas com uma redução de 86 a 91% em comparação com o tamanho original dos dados. Em um ambiente bi-dimensional, FoL foi aplicado para a segmentação de pulmão em imagens de tomografia computadorizada. O algoritmo FoL alcança bons resultados com aproximadamente 19 segundos no resultado mais significativo em um exame com 430 fatias e apresenta índices de similaridade alcançando HD 3,5, DICE 83,63, e Jaccard 99,73 e índices qualitativos alcançando Sensibilidade 83,87, MCC 83,08, e ACC 99,62. Finalmente, o FoL também foi apresentado como um transformada de agrupamento supervisionado que pode ser treinada para alcançar melhores resultados e anexado a outras abordagens como aplicações de Redes Neurais Profundas Convolutivas alcançando DB 1,74, CH 137 e Sil 0,17.

Palavras-chave: Compressão de características; algoritmo de agrupamento supervisionado; reconhecimento facial; segmentação de pulmão; aprendizado profundo

ABSTRACT

This work proposes a dimensionality-independent clustering method called *Floor of Log* (FoL) transform. The advantage of this method consists of its ease and practicality in implementation, as well as the ability to generate multiple effects on the data. Among them, clustering and compression stand out. Three applications directed to real problems were chosen. In the first, FoL was evaluated in tasks related to facial recognition, specifically on feature arrays. For this evaluation, CelebA, Extended YaleB, AR and LFW datasets were used in addition to Accuracy (ACC) and Size of post-compression datasets as evaluation metrics. The second application evaluates the FoL in a two-dimensional environment, specifically in computed tomography for lung segmentation and, consequently, within an image processing environment. In this evaluation, the LUNA16 and LAPISCO datasets were used together with the Hausdorff Distance (HD), DICE, ACC, Jaccard and Matthews Correlation Coefficient (MCC) metrics. The third application seeks to evaluate FoL in a more dimension-independent context, within general-purpose convolutional neural networks. The CIFAR10 and CIFAR100 benchmark datasets were used in addition to Davies-Bouldin (DB), Calinsky-Harabaz (CH) and Silhouette (Sil). As a result, the FoL when applied to the arrays in the CelebA, Extended YaleB, AR and LFW datasets, obtained equal or better results when compared to the approach using the same sorted with uncompressed features, but with a reduction of 86 to 91% compared to the original data size. In a two-dimensional environment, FoL was applied for lung segmentation in computed tomography images. The FoL algorithm achieves good results with approximately 19 seconds at the most significant result in a scan with 430 slices and features similarity indices reaching HD 3.5, DICE 83.63, and Jaccard 99.73 and qualitative indices reaching Sensitivity 83.87, MCC 83.08, and ACC 99.62. Finally, FoL was also presented as a supervised clustering transform that can be trained to achieve better results and attached to other approaches such as Convolutional Deep Neural Network applications reaching DB 1.74 CH 137 Sil 0.17.

Keywords: Feature compression; supervised clustering algorithm; recognition facial; lung segmentation; deep learning

LISTA DE FIGURAS

Figura 1 – Amostras de rosto selecionadas dos conjuntos de dados ExtendedYaleB, CelebA, LFW e AR.	20
Figura 2 – Sequência de Convoluções.	28
Figura 3 – Ilustração de <i>Pooling</i>	29
Figura 4 – Ilustração da aplicação do método do dropout.	30
Figura 5 – FoL aplicado a dados	38
Figura 6 – Figura com exemplos de imagens faciais de cada conjunto de dados usados: AR, Extended YaleB, LFW e CelebA, respectivamente.	40
Figura 7 – Efeito da função chão aplicada na função logarítmica com diferentes valores de bases. Em (a), algumas funções logarítmicas com bases diferentes. Em (b), a função chão aplicada ao logaritmo com diferentes valores de bases	42
Figura 8 – FoL aplicado a dados	42
Figura 9 – FoL aplicado no <i>datasets Breast, Wine</i> e Iris em diferentes bases.	43
Figura 10 – Visão geral da metodologia adotada. Primeiro, todos os conjuntos de dados faciais são carregados, seguidos pela extração de features pelo FaceNet. Na terceira etapa, a abordagem proposta é aplicada às características, reduzindo seu tamanho mantendo a qualidade. A quarta etapa apresenta a etapa de avaliação, que, neste caso, é baseada na correspondência de faces.	45
Figura 11 – Etapas do processo de clusterização. A, B e C apresenta alguns exemplos do método proposto. Após a imagem de entrada, esta foi combinada com LBP image e depois agrupados usando o algoritmo <i>Floor of Log</i> . A coluna O resultado apresenta a imagem normalizada após a operação Piso do Log.	48
Figura 12 – Operação Flood-Fill aplicada na imagem 1. O resultado foi apresentado pelo ponto 2	49
Figura 13 – A representação tridimensional da região pulmonar. Depois o processo de cada fatia, o algoritmo extrai todo o pulmão na terceira dimensão usando os 2 volumes maiores.	49
Figura 14 – Problemas na criação do dataset utilizado no processo de training. O entorno e o ROI podem ter os mesmos valores.	52
Figura 15 – O contorno verde na imagem do meio apresenta a classe negativa utilizada nesse trabalho.	52

Figura 16 – Overview do processo proposto para gerar os resultados. Inicialmente, para cada problema os datasets são coletados e posteriormente direcionados para cada aplicação. Em seguida, o FoL é aplicado em cada saída dos métodos. Finalmente, para cada aplicação os resultados específicos de cada uma serão avaliados.	57
Figura 17 – Avaliação da abordagem proposta. Para cada <i>dataset</i> de face, o FaceNet foi aplicado para extrair <i>features</i> na etapa 1. Na etapa 2, o FoL foi aplicado no dataset para cada <i>array</i> de <i>feature</i> , gerando <i>features</i> compactadas. Esses <i>datasets</i> foram salvos após esta etapa para avaliar o tamanho final do conjunto de dados. Finalmente, a etapa 3 avalia cada <i>datasetss</i> usando SVM e K-NN.	58
Figura 18 – Figura com resultados de correspondência de rostos para o <i>dataset</i> AR. Mesmo com a presença de diferentes tipos de oclusão, lenço e óculos, o FoL obtém resultados diferentes, mas mantém a mesma pessoa relacionada ao rosto.	60
Figura 19 – Comparação dos resultados das <i>features</i> compactados e descompactados considerando a precisão de correspondência facial obtida por cada classificador como métrica de avaliação no conjunto de dados LFW. Também é apresentado o tamanho de cada arquivo e a redução percentual após a compactação. . . .	62
Figura 20 – Resultados da Face Matching para o conjunto de dados Extended Yale B. Mesmo com diferentes iluminações, uma questão particular deste conjunto de dados, o FoL alcança a mesma pessoa relacionada a Face.	63
Figura 21 – Comparação do tamanho do arquivo das <i>features</i> compactados e não compactados no CelebA conjunto de dados	65
Figura 22 – Metodologia proposta. Nas etapas de aquisição de imagens, as imagens pulmonares do <i>dataset</i> de tomografia computadorizada são inicialmente armazenadas e separadas por suas classes. A coluna Pré-processamento apresenta o novo método de agrupamento proposto e a etapa de Segmentação na terceira coluna apresenta a segmentação da imagem e a construção da visualização 3D.	65
Figura 23 – Avaliação do processo de aprendizado pelo algoritmo de treinamento proposto com 10 cross-validation usando 105 iterações. A cada iteração, o algoritmo muda a base 'b' usada em FoL.	66

Figura 24 – Efeitos do FoL no <i>dataset</i> LUNA16. Os valores originais apresentados na imagem (a) foram convertidos no valor único da região aerada (b) após o método FoL.	66
Figura 25 – Processo de agrupamento no dataset LAPISCO. Após o algoritmo FoL, a região pulmonar foi extraída por limiarização a partir de um único valor de pixel.	67
Figura 26 – Exemplos de segmentação pulmonar. Após a extração dos 2 maiores volumes na combinação de todo o exame, o processo de segmentação é finalizado. . .	68
Figura 27 – Modelo geral de adição do FoL como camada dentro dos modelos.	70
Figura 28 – Processo de aprendizado do FoL em comparação com os demais métodos na rede VGG	70
Figura 29 – Processo de aprendizado do FoL em comparação com os demais métodos na rede Googlenet	71
Figura 30 – Processo de aprendizado do FoL em comparação com os demais métodos na rede Attention	71

LISTA DE TABELAS

Tabela 1 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no <i>dataset</i> AR com lenço e óculos como objetos de oclusão. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.	59
Tabela 2 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no <i>dataset</i> LFW. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.	61
Tabela 3 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no conjunto de dados Extended Yale B. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.	63
Tabela 4 – Comparison of the file size of compressed and uncompressed features in CelebA dataset.	64
Tabela 5 – Tempo médio de segmentação do FoL em relação a outros métodos.	68
Tabela 6 – Índices de similaridade com FoL em relação a outros métodos de segmentação	69
Tabela 7 – Índices quantitativos para FoL em relação a outros métodos de segmentação	69
Tabela 8 – Acurácia dos modelos no conjunto de dados CIFAR-10.	71
Tabela 9 – Acurácia dos modelos no conjunto de dados CIFAR-100.	72
Tabela 10 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando VGG16.	72
Tabela 11 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando GoogleNet.	72
Tabela 12 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando Attention.	72

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos Geral e Específicos	16
1.2	Produção científica	17
1.3	Estrutura da Tese	17
1.4	Estado da Arte	18
<i>1.4.1</i>	<i>Métodos de propósito geral</i>	<i>18</i>
<i>1.4.2</i>	<i>Métodos para compressão de dados</i>	<i>20</i>
<i>1.4.3</i>	<i>Métodos para clusterização em imagens</i>	<i>22</i>
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	Redes Neurais Artificiais	24
<i>2.1.1</i>	<i>Multilayer Perceptron</i>	<i>24</i>
<i>2.1.2</i>	<i>Convolutional Neural Networks</i>	<i>27</i>
<i>2.1.2.1</i>	<i>Convolução</i>	<i>27</i>
<i>2.1.2.2</i>	<i>Batch Normalization</i>	<i>28</i>
<i>2.1.2.3</i>	<i>Pooling</i>	<i>29</i>
<i>2.1.2.4</i>	<i>Camada Densa</i>	<i>30</i>
<i>2.1.2.5</i>	<i>Dropout</i>	<i>30</i>
<i>2.1.2.6</i>	<i>Funções de Ativação</i>	<i>31</i>
<i>2.1.3</i>	<i>Facenet</i>	<i>32</i>
<i>2.1.4</i>	<i>VGG16</i>	<i>33</i>
<i>2.1.5</i>	<i>Googlenet</i>	<i>34</i>
<i>2.1.6</i>	<i>Attention</i>	<i>36</i>
2.2	Processamento de imagens - Segmentação	37
3	METODOLOGIA	38
3.1	Datasets	39
3.2	Floor of Log	41
3.3	Análise da Complexidade Assintótica	44
3.4	Avaliação do FoL em array	44
<i>3.4.1</i>	<i>Processo de treinamento</i>	<i>46</i>
3.5	Avaliação do FoL em imagens	46

3.5.1	<i>Segmentação</i>	47
3.5.2	<i>Métricas de avaliação</i>	50
3.5.3	<i>Treinamento da base logaritmica</i>	51
3.6	Avaliação em modelos gerais	52
3.6.1	<i>Conjunto de dados</i>	53
3.6.2	<i>Métricas de avaliação</i>	53
4	RESULTADOS	56
4.0.1	<i>Configurações gerais dos parâmetros utilizados</i>	56
4.1	Avaliação do FoL em array	58
4.1.1	<i>Resultados experimentais no dataset AR</i>	59
4.1.2	<i>Resultados experimentais no dataset LFW</i>	60
4.1.3	<i>Resultados experimentais no dataset EYale B</i>	62
4.1.4	<i>Resultados experimentais no dataset CelebA</i>	63
4.2	Avaliação do FoL em Imagens	64
4.3	Avaliação do FoL dentro de modelos de Deep Learning	69
5	CONCLUSÕES E TRABALHOS FUTUROS	74
5.0.1	<i>Trabalhos Futuros</i>	74
	REFERÊNCIAS	76

1 INTRODUÇÃO

Transformar um conjunto de dados, muitas vezes, trata-se de alterar a perspectiva do mesmo, seja para mudar a visão do problema em questão, extrair novos padrões ou sintetizar dados redundantes. Algumas dessas transformadas, podem inclusive alterar a performance de uma solução, entregando dados menos ruidosos e fáceis de serem manipulados (NARGESIAN *et al.*, 2017). Por outro lado, de um ponto de vista estatístico, certas transformações mimificam a distribuição de um conjunto de dados ou mesmo alteram-a, justamente para auxiliar em uma melhor construção de novos modelos estatísticos (SHAH *et al.*, 2020), melhorando na previsibilidade. Sendo assim, as transformações tornam-se uma ferramenta importante de uma maneira geral para a ciência de dados (NIXON; AGUADO, 2019).

Mudar a perspectiva através da transformação dos dados vai além de auxiliar no desenvolvimento de novas soluções e ganho de performance uma vez que ela também contribui para o redirecionamento do foco de trabalho (TURNER *et al.*, 1999), saindo, muitas vezes, de escopos desenhados apenas em prover solução, para abordagens voltadas ao entendimento do problema em si (JACKSON, 1995).

Outro impacto da transformação de um conjunto de dados ocorre diretamente na identificação e extração de diferentes padrões (DOL; JAWANDHIYA, 2023). Atualmente, diversas aplicações, em diferentes domínios, como a educação, entretenimento, agronomia, além da avaliação e mitigação da poluição atmosférica, temos visto esse processo, no qual, muitas vezes, pode ser considerado mineração de dados, cujo ato auxilia na extração novos padrões, e que são implementados em solução de maneira abrangente e com considerável sucesso (DOMINIC *et al.*, 2023; ALLMUTTAR; ALKHAFAJI, 2023; AULAKH *et al.*, 2023; DOL; JAWANDHIYA, 2023; ZHANG *et al.*, 2023; MACUÁCUA *et al.*, 2023; XU *et al.*, 2023).

Tratando-se de transformações de dados para sintetiza-los, os estudos de Nassra e Capella (NASSRA; CAPELLA, 2023) e Pullaiah *et al.* (PULLAIAH *et al.*, 2023) destacam a importância da compressão de dados na melhoria da eficiência energética e na redução do consumo de energia em redes de sensores sem fio e sistemas integrados. Esta aplicação é particularmente relevante no contexto de cuidados de saúde e sistemas integrados em chips, onde a quantidade de dados redundantes pode ser substancial. Jha *et al.* (JHA; KOLEKAR, 2022) e Baliarsingh *et al.* (BALIARSINGH *et al.*, 2023) expandem esta aplicação para sistemas de saúde cardíaca, onde a compressão de dados do eletrocardiograma pode superar limitações de armazenamento de dados, largura de banda e vida útil da bateria.

Yemeni et al. (YEMENI *et al.*, 2021) propõem uma abordagem para reduzir a redundância de dados espaciais e temporais, com o objetivo de prolongar a vida útil das redes de sensores sem fio. A mesma ideia de redução de redundância é usada por Xu et al. (XU *et al.*, 2021) para reduzir o esforço de anotação na identificação de pessoas, destacando a aplicação em diversos domínios. Além disso, Wang et al. (WANG *et al.*, 2016) e He et al. (HE *et al.*, 2023) usam o aprendizado de representação para reduzir a redundância de dados espaciais e temporais e a redundância em redes neurais de grafos, respectivamente, mostrando que a ideia de redução de redundância pode ser implementada em diferentes níveis do processamento de dados.

Dito isso, encontramos na transformação de dados a necessidade constante de identificação de padrões, segmentação e sintetização, tornando-se convidativo a busca por soluções que unam estas funcionalidades. A transformada *Floor of Log* (FoL) é, neste sentido, uma ferramenta capaz de causar impacto nesses domínios, mesmo não tendo sua função primária designada para tais ações. A aplicação de sua fórmula pode ser verificada em dados unidimensionais, bidimensionais ou de maior dimensionalidade, encontrando resultados com efeitos de clusterização, compressão de dados e até mesmo segmentação de imagens, materializando a versatilidade da transformada FoL.

Além disso, a alteração da FoL, através dos parâmetros "bias" e "base", confere uma adaptabilidade, possibilitando um alinhamento mais coerente ao problema em estudo. Esta flexibilidade se une à capacidade matemática da FoL de promover efeitos em um único procedimento, sugerindo como um instrumento interessante na manipulação de dados. De modo conciso, a FoL busca atender à demanda por ferramentas versáteis e eficazes na ciência de dados, na tentativa de auxiliar demais métodos junto ao domínio científico.

1.1 Objetivos Geral e Específicos

Esse trabalho tem como objetivo geral estabelecer uma comparação do FoL em diversas áreas de aplicação, particularmente em otimização, processamento de imagens e modelos de aprendizado profundo e, conseqüentemente, avaliar a performance do FoL nestes diferentes cenários de aplicação além de analisar seu impacto na melhoria da eficiência computacional.

Neste sentido, este trabalho concentra-se em:

- Aplicar o FoL em arrays unidimensionais obtidos a partir da saída da Facenet para quatro *datasets* distintos. O objetivo é avaliar o desempenho do FoL na otimização dessas representações vetoriais.

- Aplicar o FoL em imagens médicas bidimensionais, especificamente em unidades Hounsfield (HU), com o objetivo de aferir a eficácia do FoL na análise e otimização do processamento dessas imagens.
- Avaliar o FoL na otimização de características multidimensionais extraídas das Convolutional Neural Networks (CNNs) com e sem pré-treinamento: Facenet, VGG16, Googlenet e Attention. A meta é entender como o FoL pode influenciar e otimizar a performance do produto dessas redes.

1.2 Produção científica

Durante este projeto de doutorado, os seguintes trabalhos científicos foram aceitos e publicados, a saber:

- **Peixoto, Solon Alves**, et al, Medeiros, A. G., Hassan, M. M., Dewan, M. A. A., Albuquerque, V. H. C. D., Filho, P. P. R. (2020). Floor of log: a novel intelligent algorithm for 3D lung segmentation in computer tomography images. *Multimedia Systems*, 1-13
- **Peixoto, Solon Alves**, Vasconcelos, F. F., Guimarães, M. T., Medeiros, A. G., Rego, P. A., Neto, A. V. L., ... Rebouças Filho, P. P. (2020). A high-efficiency energy and storage approach for IoT applications of facial recognition. *Image and Vision Computing*, 96, 103899..
- Rebouças, Elizangela de Souza; De Medeiros, Fátima Nelsizeuma Sombra ; Marques, Regis Cristiano P. ; Chagas, João Vitor S. ; Guimarães, Matheus T. ; Santos, Lucas Oliveira. ; Medeiros, Aldisio G. ; **Peixoto, Solon Alves**. Level set approach based on Parzen Window and floor of log for edge computing object segmentation in digital images. *Applied Soft Computing*, v. 105, p. 107273, 2021.
- Albuquerque, Victor Hugo C. ; de A. Rodrigues, Douglas ; Ivo, Roberto F. ; **Peixoto, Solon Alves**. Fast fully automatic heart fat segmentation in computed tomography datasets. *Computerized Medical Imaging and Graphics*, v. 80, p. 101674, 2020.

1.3 Estrutura da Tese

Este trabalho apresenta na Seção 1.4 o Estado da Arte de diversas transformadas e suas aplicações e devida importância em cada setor. O Capítulo 2 traz consigo os principais conceitos utilizados nesse trabalho, principalmente associados às aplicações. O Capítulo 3 apresenta

a metodologia utilizada para obter os resultados incluindo a apresentação da transformada FoL e seu detalhamento como, também, o comportamento da transformada em aplicações distintas. Os resultados são apresentados e discutidos no Capítulo 4, comparando com outras configurações.

1.4 Estado da Arte

Na área das transformadas, de uma forma geral, encontramos uma constante mudança e progresso. As transformadas são ferramentas úteis e são aplicadas em muitas áreas diferentes, como o processamento de sinais, a análise de dados e a computação gráfica. No entanto, mesmo sendo tão úteis, a maneira como usamos as transformadas está sempre mudando e evoluindo através de novas técnicas e abordagens que estão sendo descobertas, e isso torna o estudo das transformadas um campo interessante.

O domínio da transformação de dados é distribuído por várias abordagens desde os métodos de transformação de probabilidade condicional sintonizados, até técnicas evolutivas para melhorar a previsão prognóstica. Isso reflete, em outras palavras, que embora avanços significativos tenham sido feitos, existem buscas mais objetivas do ponto de vista estatístico enquanto outras mais focadas em técnicas de aproximação.

1.4.1 Métodos de propósito geral

Li et al. (LI *et al.*, 2019) propuseram o método *Fine-Tuned Conditional Probability Transformation* (FTCPT) para a transformação de atributos nominais, um passo crítico na preparação de dados para muitos algoritmos de aprendizado de máquina. Já Parisi e RaviChandran (PARISI; RAVICHANDRAN, 2020) visam melhorar a previsão prognóstica da hepatite através da transformação evolutiva de características. Ambos exemplificam a ampla aplicabilidade da transformação de dados, seja na melhoria do desempenho dos algoritmos de aprendizado de máquina ou na melhoria da precisão de importantes previsões clínicas.

Os desafios permanecem em termos de entender a importância da transformação de características para modelos específicos, como os modelos de classificação neural (ZHUANG *et al.*, 2020). Além disso, a busca por uma transformação de recursos auto-otimizada (XIAO *et al.*, 2022), e a combinação de algoritmos binários de *swarm* com métodos híbridos de transformação de dados (SHEKHAWAT *et al.*, 2021), demonstram a constante procura por soluções mais eficazes. Essa busca por abordagens mais avançadas, mostra que o campo ainda tem muitas

direções e oportunidades a serem exploradas.

O estado da arte em agrupamento de dados evoluiu significativamente nos últimos anos, abordando várias perspectivas e metodologias com a finalidade de proporcionar agrupamentos mais eficazes do ponto de vista de coesão interna e externa. A concepção dessa área complexa é refletida na diversidade de abordagens, com destaque para os métodos baseados em grafos, algoritmos genéticos e análises estatísticas.

A agrupamento de dados baseado em grafos é uma das abordagens pioneiras e mais robustas na área. Liu e Barahona (2020) (LIU; BARAHONA, 2020) propuseram uma abordagem de agrupamento de dados não supervisionado baseado em comunidade multiescala, utilizando o *Markov Stability*, um framework baseado em dinâmica, para realizar a detecção de comunidades em diferentes níveis de resolução. Este método foi aplicado com sucesso em diversos contextos, incluindo análise de redes neurais, estruturas de proteínas e redes sociais. Contudo este método, embora promissor, ainda enfrenta desafios em termos de otimização da construção do gráfico e na avaliação da estrutura de dados apropriada.

Os algoritmos genéticos híbridos representam uma nova vertente no panorama do agrupamento de dados, oferecendo soluções otimizadas para os problemas enfrentados pelos métodos tradicionais como *k-means* e *k-medoids*. Islam et al. (2019) (ISLAM *et al.*, 2019) propuseram um algoritmo genético híbrido (HGA) para melhorar a precisão do agrupamento. Este método baseia-se na utilização de técnicas de *k-means* e *k-medoids* para determinar os centros de agrupamento, e na aplicação de um algoritmo genético para evitar a otimização local. Este método demonstrou uma melhoria de 2,67% a 28,68% na precisão do agrupamento em relação aos métodos tradicionais. No entanto, o escopo para a exploração dos algoritmos genéticos no agrupamento de dados ainda é vasto.

Finalmente, a análise estatística de agrupamento de dados representa um passo adiante na obtenção de resultados confiáveis e significativos. Mirzal (MIRZAL, 2020) realizou uma análise estatística da performance de agrupamento entre diversos métodos, incluindo NMF, agrupamento espectral, *Gaussian Mixture Model* e *k-means*. Este estudo sugere que a análise estatística pode fornecer uma avaliação robusta da performance de diferentes métodos de agrupamento, ao mesmo tempo que destaca a necessidade de considerar a estrutura dos dados ao projetar algoritmos de agrupamento. Ainda assim, a adoção de análises estatísticas na avaliação de performance de agrupamento ainda é subutilizada, apontando para uma área potencial de crescimento futuro nesta área.

Figura 1 – Amostras de rosto selecionadas dos conjuntos de dados ExtendedYaleB, CelebA, LFW e AR.



Fonte: Elaborada pelo Autor

1.4.2 Métodos para compressão de dados

Com a crescente geração de dados no cenário atual, conhecido como *Big Data*, a compressão de dados se tornou um campo de estudo fundamental. Sua importância, está no fato de que a compressão de dados pode auxiliar na economia de armazenamento, pode otimizar o uso da largura de banda, e melhorar a velocidade de transmissão de dados. Com a compressão de dados eficaz, lidar com os desafios que vem junto ao manuseio de volumes grandes de dados, se torna mais gerenciável e útil.

Devido ao rápido crescimento contínuo das tecnologias de reconhecimento facial, e a sua integração em sistemas como a Internet das Coisas, ou *IoT*, esta área de aplicação se tornou especialmente relevante. Os dados gerados por esses sistemas, são de volume e complexidade grandes, trazendo desafios específicos que podem ser efetivamente abordados, através de técnicas de compressão de dados adaptativas. Então, serão explorados mais profundamente essas técnicas, com um foco especial na aplicação delas, em sistemas de reconhecimento facial.

Nos últimos anos, pesquisadores têm se concentrado não apenas no reconhecimento facial, mas também na identificação de emoções (KAYA *et al.*, 2017), expressões faciais (ZENG *et al.*, 2018), e até mesmo idade e gênero (FANG *et al.*, 2019). Nesse sentido, alguns conjuntos de dados foram criados para explorar esses diferentes cenários, conforme mostrado na Figura 1.

Essas pesquisas demonstram uma evolução significativa nos métodos de detecção, identificação e reconhecimento facial. No entanto, a etapa de extração das redes de aprendizado

profundo pode gerar milhares de características, que impactam em arquivos com grande volume de dados, dificultando o uso dessas técnicas em sistemas embarcados ou até mesmo smartphones.

Em associação com a evolução das técnicas de reconhecimento facial (GUIMARAES *et al.*, 2019), consideráveis pesquisas têm se concentrado em integrar essas técnicas com sistemas e dispositivos da Internet das Coisas (MEHMOOD *et al.*, 2019; ULLAH *et al.*, 2019; SODHRO *et al.*, 2019b). Os sistemas *IoT* são capazes de interconectar vários objetos físicos pela internet, coletando, transmitindo e processando dados entre esses dispositivos (HUSSAIN *et al.*, 2019a).

Para estabelecer a comunicação entre os dispositivos *IoT*, eles devem ter a capacidade de entender a linguagem do dispositivo com o qual estão se comunicando. Devido ao crescente número de dispositivos *IoT*, são criadas mais formas de comunicação entre tais dispositivos, o que pode dificultar a troca de informações entre dispositivos que não fazem parte de uma solução integrada (CRUZ *et al.*, 2018). Portanto, algumas pesquisas se concentraram em plataformas de *middleware* (NGU *et al.*, 2016; FARAHZADI *et al.*, 2018), ou seja, plataformas capazes de integrar sistemas *IoT*, permitindo que eles se comuniquem e tomem decisões com base nos dados coletados de cada dispositivo (CRUZ *et al.*, 2018).

Além disso, os dispositivos *IoT* geralmente transmitem um volume considerável de dados, então eles precisam de uma conexão rápida com a internet e consomem grandes quantidades de energia ao enviar dados (AZAR *et al.*, 2019; RANDHAWA *et al.*, 2019; HUSSAIN *et al.*, 2019b). Portanto, é de fundamental importância para a integração dos métodos de aprendizado profundo com os dispositivos *IoT* para reconhecimento facial que as características geradas por essas redes neurais complexas sejam comprimidas em pacotes de dados menores sem afetar significativamente a precisão do método.

Azar *et al.* (AZAR *et al.*, 2019) propõem uma abordagem para reduzir o tráfego de dados usando compressão *lossy* limitada por erro. Em seu trabalho, técnicas de aprendizado de máquina supervisionado realizam a reconstrução dos dados. Essa abordagem foi testada na coleta de dados de saúde usando uma rede de sensores corporais sem fio, onde a compressão alcançou até 103x sem afetar a qualidade dos dados.

No trabalho de Al-Maitah *et al.* (AL-MAITAH *et al.*, 2019), uma técnica chamada Balanço de Memória Seletiva permite melhor gerenciamento de memória para dispositivos de atuação. Essa técnica foi construída a partir de modelos de aprendizado de máquina sequenciais, onde análises periódicas são feitas para melhor gerenciamento desse armazenamento. Nesse sentido, a memória pode ser gerenciada de forma adaptativa, minimizando o espaço alocado.

Mukherjee et al. (MUKHERJEE *et al.*, 2019) propõem uma redução de 93% no consumo de energia de dispositivos móveis que executam aplicativos de reconhecimento facial. A técnica apresentada é dividida em duas etapas em que o dispositivo móvel primeiro detecta o rosto e envia apenas a região de interesse para um servidor na nuvem, usado para o reconhecimento facial na segunda etapa.

Moon et al. (MOON *et al.*, 2018) fizeram uma avaliação importante entre diferentes métodos de compressão aplicados a dados climáticos. Seu trabalho mostrou uma relação entre o nível de compressão de dados e a discriminabilidade desses dados após a compressão. Outro fator importante destacado foi a granularidade, que pode oferecer vantagens em problemas de previsão, bem como flexibilidade no processo de compressão.

1.4.3 Métodos para clusterização em imagens

Alguns sinais de que o paciente tem a presença de patologias associadas, por exemplo, ao pulmão incluem sinusite crônica, degradação do movimento elástico dos pulmões, tosse, fadiga e fraqueza. A gravidade dessas patologias requer métodos rápidos que aumentem o desempenho do prognóstico e o tratamento do paciente. Além disso, uma alta demanda de pacientes implica em métodos necessários que ajudem na assistência aos diagnósticos clínicos, evitando a possibilidade de erros decorrentes de fatores fisiológicos humanos (FÉLIX, 2011) e outras tecnologias capazes de interconectar tais informações (SODHRO *et al.*, 2019b; SODHRO *et al.*, 2019a; GUMAEI *et al.*, 2019).

Recentemente, algoritmos baseados em aprendizado profundo têm a capacidade de lidar com enormes quantidades de dados de saúde multimídia e são capazes de extrair dados suficientes para sua análise e foram criados para atender a essa exigência em diferentes cenários médicos (RODRIGUES *et al.*, 2018; SODHRO *et al.*, 2018; SODHRO *et al.*, 2016; SODHRO *et al.*, 2017; CHOUHAN *et al.*, 2020; REBOUÇAS *et al.*, 2019; GUPTA *et al.*, 2019; HASSAN *et al.*, 2019). Alguns desses métodos se concentram na detecção de nódulos na região pulmonar (RODRIGUES *et al.*, 2018), e a análise de patologias pulmonares (RODRIGUES *et al.*, 2018), bem como o estudo em outras doenças como prognósticos de Parkinson (SOUZA *et al.*, 2018).

Em tarefas de bioimagem relacionadas à segmentação pulmonar, a detecção da região pulmonar é essencial e tomografia computadorizada (TC) e ressonância magnética (RM) são ferramentas importantes para obtê-las (HAN *et al.*, 2019). Essas tarefas de detecção normalmente estão relacionadas ao conhecimento a priori, neste caso, as características de uma lesão, bem

como sua forma e delimitação, são baseadas no conhecimento de um especialista (FANG *et al.*, 2017).

Maier et al. (2017) (MAIER *et al.*, 2017) apresenta uma referência com comparação de algoritmo relacionado a tarefas de detecção de Acidente Vascular Cerebral (AVC) isquêmico em exames de RM. Em seu trabalho, o desenvolvimento de abordagens que combinam métodos para enfatizar diferentes características apresenta melhores resultados, onde as redes neurais convolucionais (CNN) são a abordagem mais comum para extrair múltiplos padrões.

Rebouças Filho et al. (2017) (REBOUÇAS *et al.*, 2017b) apresentou um poderoso método usando o Método de Contorno Ativo chamado Método de Contorno Ativo Crisp Adaptativo (ACACM). Os resultados mostraram tanto a velocidade quanto a eficácia contra softwares comerciais. Uma das combinações mais importantes deste método, é a inclusão do Perceptron MultiLayer no algoritmo, tornando a forma adaptável à imagem com alguma inteligência (HAJIMANI *et al.*, 2017).

Nithila e Kumar (2019) (NITHILA; KUMAR, 2019) também apresentam um método baseado no modelo de Contorno Ativo chamado *Selective Binary and Gaussian Filtering-new Signed Pressure Force*, onde ele pode detectar as bordas do parênquima dos pulmões com muito mais eficiência, destacando a diferença entre o lado interno e externo dos pulmões.

Zhang et al. (2018) (ZHANG *et al.*, 2018) apresenta um algoritmo GrabCut aprimorado para segmentação do parênquima pulmonar. A principal contribuição deste trabalho está em otimizar as etapas selecionando automaticamente o parênquima pulmonar que pode ser obtido usando um algoritmo de limiarização OTSU.

Outros campos importantes de trabalhos relacionados estão associados à segmentação de nódulos pulmonares. No trabalho apresentado por Shakibapour et al. (SHAKIBAPOUR *et al.*, 2019) uma abordagem não supervisionada baseada em meta-heurísticas é usada para a segmentação de nódulos pulmonares em tomografias de pulmão. Shakir et al. (SHAKIR *et al.*, 2018) estende a segmentação do nódulo para uma abordagem de segmentação 3d baseada no contorno ativo geodésico.

Redes neurais profundas também são uma ferramenta poderosa para o processo de segmentação (YANG *et al.*, 2016). Skourt et al. (SKOURT *et al.*, 2018) usam a arquitetura U-net para comprimir e extrair as informações dos dados. Este processo permite que o modelo aprenda as informações mais importantes relacionadas ao pulmão e depois use essas informações para segmentar outros pulmões.

2 FUNDAMENTAÇÃO TEÓRICA

A avaliação da transformada FoL será aplicada sobre tipos de dados distintos, arrays de features, imagens médicas e mapas de características oriundas de CNNs, que por sua vez possuem diferenças tanto na sua estrutura quanto na sua representação. Nesse sentido, esta seção fornece o contexto necessário para a avaliação do FoL nesses diferentes cenários.

2.1 Redes Neurais Artificiais

Muito utilizadas atualmente, as redes neurais artificiais (RNA) são ferramentas, ou modelos, que "mimificam" o comportamento do cérebro humano e se dedicam a aprender e a reconhecer padrões (HAYKIN, 2009). São constituídas por unidades de processamento, ou 'neurônios', dispostos em camadas que se interligam através de conexões ponderadas, viabilizando o aprendizado não supervisionado.

As RNAs possuem ampla aplicação, incluindo reconhecimento de padrões, aprendizado de máquina, processamento de linguagem natural, entre outros (SCHMIDHUBER, 2015). As camadas ocultas permitem a modelagem de problemas não lineares complexos, tornando as RNAs ferramentas poderosas para diversos desafios em ciência de dados (GOODFELLOW *et al.*, 2016). As arquiteturas de redes neurais incluem Perceptron de múltiplas camadas, redes de Hopfield, máquinas de Boltzmann, redes neurais convolucionais, redes neurais recorrentes, redes adversárias generativas, entre outras, cada uma adequada a diferentes tipos de problemas (LECUN *et al.*, 2015).

Resumindo, as Redes Neurais Artificiais oferecem um *framework* robusto e flexível para a modelagem e resolução de uma variedade de problemas complexos de aprendizado de máquina.

2.1.1 *Multilayer Perceptron*

O Perceptron de Múltiplas Camadas (MLP, de "MultiLayer Perceptron" em inglês) é uma das estruturas de Redes Neurais Artificiais mais empregadas, sendo considerado uma evolução natural do Perceptron Simples. A principal limitação do Perceptron Simples é sua capacidade de aprender somente funções que são separáveis linearmente, enquanto os MLPs têm a habilidade de aprender funções não-linearmente separáveis, o que os torna bastante robustos e flexíveis (HAYKIN, 2009). Uma representação da saída de uma camada pode ser vista na

Equação 2.1.

$$\hat{y} = \phi \left(\sum_{i=1}^p x_i w_i + b \right) \quad (2.1)$$

onde ϕ corresponde à função de ativação, x_i é o vetor de entrada e w_i é o vetor de pesos que modula a entrada, ambos possuindo a mesma dimensão p . O termo de viés b bem como o vetor w_i são parâmetros ajustados durante o processo de treinamento do Perceptron. Este treinamento ocorre de forma iterativa, como indicado pela Equação 2.2,

$$w_i(t+1) = w_i(t) + \eta(y - \hat{y}(t))x_i, \quad \forall i = 1, \dots, p \quad (2.2)$$

onde t representa a iteração de treinamento, y é a classe da amostra e $\hat{y}(t)$ é a saída obtida para esta amostra na respectiva iteração. Ademais, η é a taxa de aprendizado, que tem como função ponderar o erro de previsão. Conscientes de que o Perceptron não conseguia resolver problemas não lineares, o MLP surgiu como uma configuração de Perceptrons, agora denominados neurônios, organizados em camadas ligadas de forma progressiva, que empregam uma função de ativação sigmóide, a qual é encarregada de introduzir a não linearidade no vetor de características. Cada neurônio tem a sua saída determinada pela camada na qual se encontra, conforme definido na Equação 2.3,

$$\hat{y}_{l_k} = \begin{cases} \phi_{0_k} \left(\sum_{i=1}^p x_i w_{0_{k_i}} + b_{0_k} \right) & \text{se } l = 0 \\ \phi_{l_k} \left(\sum_{i=1}^D \hat{y}_{l-1} w_{l_{k_i}} + b_{l_k} \right) & \text{se } l \neq 0 \end{cases} \quad (2.3)$$

onde l representa o número da camada e k é a posição do neurônio nessa camada, sendo x um vetor de atributos de dimensão p . \hat{y}_{l-1} representa a saída da camada anterior. Quando $l = 0$, o número de neurônios é igual à dimensão de x e, quando $l \neq 0$, o número de neurônios é D . Além disso, ϕ é a função de ativação não linear. O treinamento de um MLP normalmente consiste na propagação direta dos dados de entrada pela rede para produzir uma saída, seguida pela retropropagação do erro pela rede para atualizar os pesos. Esse processo é conhecido como *backpropagation* e é comumente realizado em conjunto com um algoritmo de otimização, como o Descida de Gradiente (RUMELHART *et al.*, 1986). Para tal, é necessário recorrer à Equação 2.4 para que esses pesos possam ser empregados,

$$\hat{w}_{l_{k_i}}(t+1) = \begin{cases} w_{0_{k_i}}(t) + \eta \cdot \delta_{l_i}(t) \cdot x_k(t) & \text{se } l = 0 \\ w_{l_{k_i}}(t) + \eta \cdot \delta_{(l+1)_i}(t) \cdot \hat{y}_{l_k}(t) & \text{se } l \neq 0 \end{cases} \quad (2.4)$$

na qual $\delta_{l_k}(t)$ é o gradiente local na camada l -ésima do k -ésimo neurônio, e pode ser expresso pela Equação 2.5:

$$\delta_{l_k} = \begin{cases} \phi'_{l_k}[u_{l_k}(t)] \frac{1}{2M} \sum_{m=1}^M [y_{l_m} - \hat{y}_{l_m}(t)]^2 & \text{se } l = z \\ \phi'_{l_k}[u_{l_k}(t)] \sum_{j=1}^J w_{l_{k_i}}(t) - \delta_{(l+1)_j}(t) & \text{se } l \neq 0 \end{cases} \quad (2.5)$$

onde z representa a camada de saída da rede, que possui M neurônios, e l é qualquer outra camada com J neurônios. Além disso, ϕ' é a derivada da função de ativação empregada, que pode ser sigmoide ou tangente hiperbólica, e sua derivada será expressa conforme a Equação 2.6,

$$\phi'_{l_k}[u_{l_k}(t)] = \begin{cases} y_{l_k}(t)[1 - y_{l_k}(t)] & \text{se } \textit{logística ou sigmoide} \\ \frac{1 - y_{l_k}^2(t)}{2} & \text{se } \textit{tangente hiperbólica} \end{cases} \quad (2.6)$$

Um MLP é composto por pelo menos três camadas de neurônios: uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada neurônio numa camada é conectado a todos os neurônios na camada seguinte, fazendo do MLP uma rede "completamente conectada" ou "densa". Os neurônios em um MLP são normalmente não-lineares, com a função de ativação sigmoideal ou tanh sendo as mais comuns. Contudo, funções de ativação mais modernas como /textitRectified Linear Unit (ReLU) são frequentemente usadas por sua eficiência e melhor desempenho em muitos casos (GOODFELLOW *et al.*, 2016).

Um desafio no treinamento de MLPs é a possibilidade de *overfitting*, que ocorre quando a rede aprende os dados de treinamento além do necessário, ao ponto de adaptar o aprendizado as amostras, reduzindo seu potencial de generalização para amostras não conhecidas durante o treinamento. Técnicas de regularização como *dropout* ou *early stopping* podem ser usadas para mitigar este problema (SRIVASTAVA *et al.*, 2014a).

Os MLPs têm uma ampla gama de aplicações, incluindo reconhecimento de padrões, previsão de séries temporais, classificação, regressão e muito mais. Embora outras arquiteturas de redes neurais, como Redes Neurais Convolucionais e Redes Neurais Recorrentes, possam superar os MLPs em tarefas específicas, os MLPs continuam sendo uma ferramenta importante e útil em aprendizado de máquina.

Apesar das vantagens dos MLPs, eles têm suas limitações. Uma delas é o problema de *vanishing gradients*, onde os gradientes se tornam muito pequenos à medida que são propagados de volta através das camadas da rede durante o treinamento. Isso torna o treinamento de MLPs profundos (i.e., com muitas camadas ocultas) desafiador. Novas técnicas e arquiteturas foram propostas para resolver este problema, como a inicialização de Xavier e as unidades de normalização em lote (HE *et al.*, 2015; IOFFE; SZEGEDY, 2015).

Em resumo, os MLPs são uma arquitetura de rede neural fundamental e versátil que é capaz de aprender representações complexas e realizar uma ampla gama de tarefas de aprendizado de máquina. Apesar de suas limitações, os avanços recentes na pesquisa de redes neurais continuam a expandir as capacidades e eficácia dos MLPs (GOODFELLOW *et al.*, 2016).

2.1.2 Convolutional Neural Networks

A Rede Neural Convolutiva (CNN) é um tipo de estrutura usada em aprendizado de máquina profundo. Esta rede usa um algoritmo para processar dados de entrada, dando relevância a diferentes características por meio da atribuição de pesos específicos. Os procedimentos, as operações e as funções de ativação que são empregadas na construção, treinamento e validação de uma CNN serão detalhados nesta seção. Nesta seção, o objetivo é facilitar o entendimento do funcionamento dessas redes complexas.

2.1.2.1 Convolução

Convolução é a aplicação de um filtro com o propósito de extrair o maior número de informações relevantes das imagens ou dados de entrada ou de camadas anteriores (ALBAWI *et al.*, 2017). Conforme a arquitetura da CNN, diferentes filtros são aplicados com o objetivo de extrair características como borda, contraste, textura, gradiente e outros, quando aplicados em imagens.

Na convolução do filtro K em uma imagem I , o filtro percorre a imagem por completo. Quando existe sobreposição entre ambos, é realizado um somatório dos produtos de elementos correspondentes, conforme a Equação 2.7.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (2.7)$$

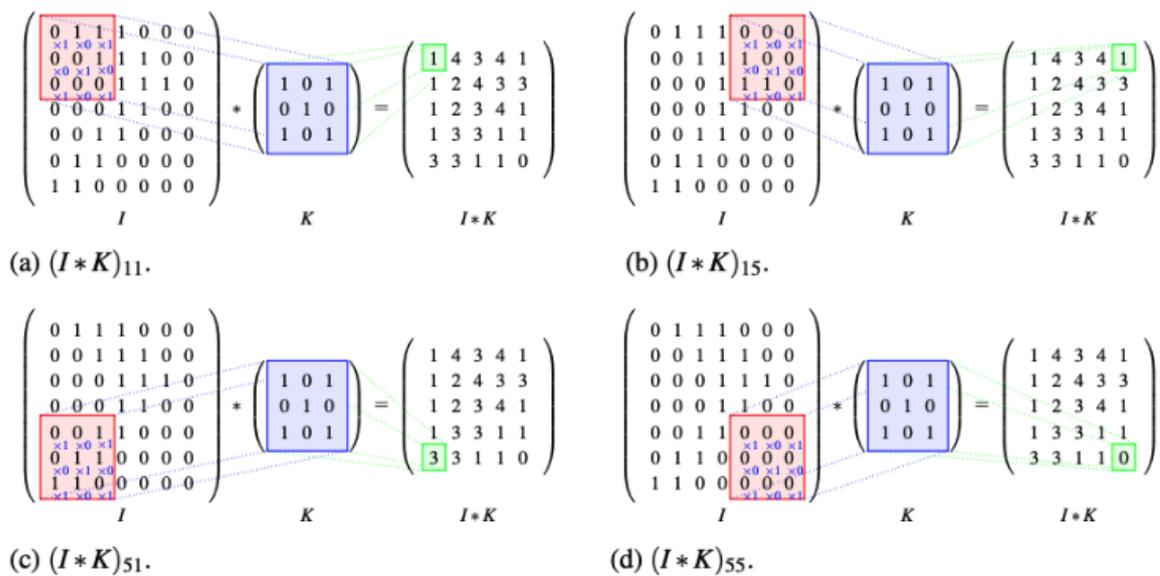
onde S é o mapa de características gerado pela convolução do filtro K na imagem I . Na Figura

2.7 é ilustrado o fluxo das quatro iterações da convolução, no exemplo, um filtro de dimensão 3×3 aplicado a uma matriz de dimensão 7×7 , sem *padding* e com *stride*, passo no qual o filtro se desloca. A Equação 2.8 representa a ordem da matriz de saída, G .

$$G = \frac{N - F}{S} + 1 \quad (2.8)$$

na qual N é a ordem da matriz, F é a ordem do filtro da convolução e S é o tamanho do *stride*. Na Figura 2a observa-se a iteração que resulta no elemento $(1, 1)$ da matriz resultante, através do somatório dos produtos de elemento por elemento do filtro com a matriz de entrada. Já na Figura 2b apresenta a iteração que resulta no elemento $(1, 5)$ da matriz de saída IK . Enquanto as Figuras 2c e 2d ilustram a mesma operação para os elementos $(5, 1)$ e $(5, 5)$ de IK , respectivamente.

Figura 2 – Sequência de Convoluções.



Fonte: Elaborada pelo autor.

2.1.2.2 Batch Normalization

Uma limitação presente no treinamento de redes neurais de aprendizado profundo é que a distribuição das entradas de cada camada se altera durante o treinamento à medida que os parâmetros das camadas anteriores mudam. Isso impacta na velocidade do treinamento (IOFFE; SZEGEDY, 2015).

Proposta por Ioffe e Szegedy (2015), a *Batch Normalization* é um método de otimização utilizado para acelerar e estabilizar treinamentos de redes neurais profundas. Nesse

processo, ocorre uma normalização das entradas de uma camada por lote, no qual a rede é dividida previamente. Esta normalização é feita pela média e desvio padrão, conforme descrito na Equação 2.9:

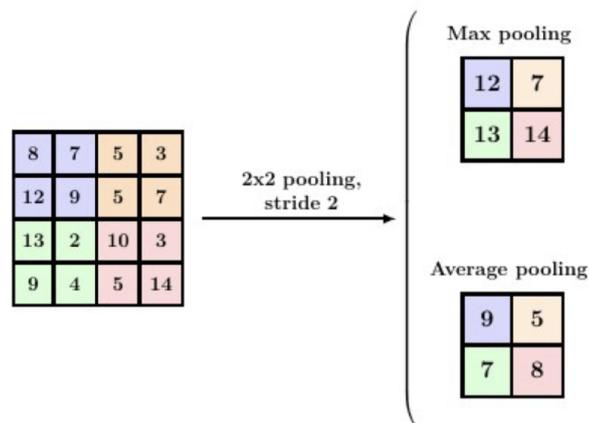
$$H' = \frac{H - \mu}{\sigma} \quad (2.9)$$

Com isso, ao estabilizar o processo de aprendizado da rede neural, o *Batch Normalization* reduz o número de épocas necessárias para a etapa de treinamento, uma vez que os pesos são atualizados ao final do processamento de cada lote e não da rede completa como é feito tradicionalmente.

2.1.2.3 Pooling

Em algumas situações, a dimensão dos dados de entrada na CNN pode interferir no seu funcionamento. Para contornar esse problema, aplica-se a operação de *pooling* na saída da camada convolucional. Este método usa campos receptivos sobre a matriz de entrada para reduzir a quantidade de informação, diminuindo assim a dimensão desses dados (SCHERER *et al.*, 2010).

Figura 3 – Ilustração de *Pooling*.



Fonte: Elaborada pelo autor.

Na Figura 3 são ilustradas as duas formas mais usadas de pooling, *max pooling* e *average pooling*, onde os filtros receptivos possuem tamanho 2x2 e percorrem a matriz 4x4. No fim do processo, há a redução da dimensionalidade da matriz. No exemplo, a matriz resultante é de 2x2 e os valores são de acordo com a operação de *pooling* aplicada.

A operação de *max pooling* seleciona o maior valor do campo receptivo e seu valor é atribuído à matriz de saída. Na operação de *average pooling* o valor atribuído à matriz de saída é o resultado do cálculo da média aritmética dos valores presentes no campo receptivo (GOODFELLOW *et al.*, 2016).

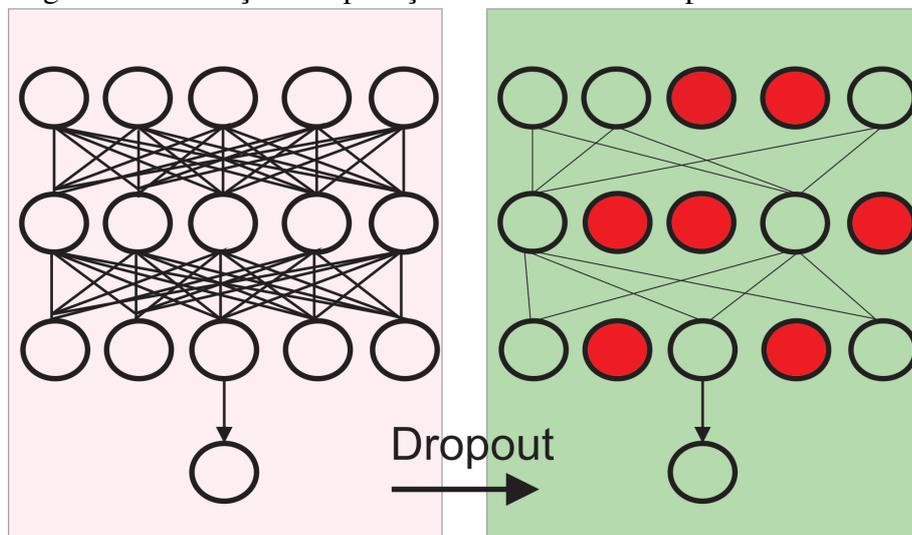
2.1.2.4 Camada Densa

Camada densa ou totalmente conectada é um conjunto de neurônios que operam propagando adiante as informações que pondera na sua entrada. Quando posicionada na saída da rede, ela será a camada responsável pela classificação do vetor de características ao final das operações de convolução e *pooling* dentro de uma rede neural (LECUN *et al.*, 2015). Ao longo de uma rede neural os dados de entrada são filtrados através das convoluções e tem a dimensão reduzida após a operação de *pooling*, ao final, no topo da rede, antes da camada densa, apenas um vetor de características restará, e a partir dele ocorrerá a predição, dentre as classes existentes no conjunto de treino.

2.1.2.5 Dropout

Proposto por Srivastava *et al.* (2014b), o dropout é uma técnica utilizada na regularização de uma rede neural. Especificamente, essa técnica é aplicada para evitar o sobreajuste do aprendizado de uma rede neural no subconjunto de treinamento, também chamado de *overfitting*. A Figura 4 ilustra o funcionamento do método.

Figura 4 – Ilustração da aplicação do método do dropout.



Fonte: Elaborada pelo Autor

O método desativa uma taxa fixa de neurônios em cada camada aplicada. No exemplo acima, dos cinco neurônios dois estão desativados, representados pela cor vermelha, o que representa 40% dos neurônios. Isso significa que 40% dos neurônios não contribuem na alimentação direta, assim como não têm pesos atualizados pelo algoritmo de *backpropagation*.

2.1.2.6 Funções de Ativação

A função de ativação desempenha o papel de determinar o limiar de decisão no espaço de entrada, estabelecendo uma fronteira no campo induzido localmente. Na ausência de uma função de ativação, o sinal de saída se reduz a uma simples função linear. As estruturas de deep learning utilizam funções de ativação para executar uma variedade de cálculos entre as camadas ocultas e as camadas de saída da arquitetura (SHEN *et al.*, 2017). As funções de ativação mais utilizadas nas redes neurais são:

- **Função de Ativação Sigmoid:** A função sigmoid é uma função de ativação que retorna um valor de probabilidade entre 0 e 1. A principal vantagem da função sigmoid é que ela existe entre (0 a 1). Portanto, ela é especialmente usada para modelos onde temos que prever a probabilidade como uma saída. A função é diferenciável, o que significa que podemos encontrar a inclinação da curva sigmoid em dois pontos dados.
- **Função de Ativação ReLU (Unidade Linear Retificada):** A função ReLU é a função de ativação mais usada nas redes neurais e deep learning. Como o nome sugere, esta função substitui todos os valores negativos na matriz de entrada por zero. Essa função acelera a convergência do SGD em comparação com a função sigmoid / tanh. Se a entrada é negativa, ela será convertida para zero e permanece a mesma se a entrada for positiva.
- **Função de Ativação TanH:** A função TanH, ou tangente hiperbólica, é a função de ativação que varia de -1 a +1. A função é diferenciável. A função TanH é melhor que a função de ativação sigmoide porque resolve o problema de vazamento de neurônio, pois os valores de saída variam de -1 a +1, portanto, a média dos valores de saída será 0.
- **Função de Ativação Softmax:** A função softmax é uma função de ativação que normaliza uma lista de números em uma lista de valores entre 0 e 1 que somam 1. A função é útil na saída de rede de sistemas de classificação multiclasse.

A escolha da função de ativação depende do tipo de tarefa de aprendizado de máquina que está sendo implementada e dos dados de entrada. As funções de ativação adicionam não linearidade ao modelo, o que ajuda a capturar relações mais complexas entre as variáveis de

entrada e saída. Elas também ajudam a controlar a saída da rede neural, mantendo-a em um intervalo específico.

2.1.3 *Facenet*

Schroff et. al (SCHROFF *et al.*, 2015) apresenta uma proposta chamada FaceNet que aprende diretamente um mapeamento de imagens de rosto para poder ser aplicados em tarefas relacionadas a verificação de rosto, ou *face matching*. Nesse caso, esse mapeamento chega ao final da rede em um formato de array, ou encodings, que por sua vez podem ser comparados entre outros encodings por algum critério de distância. O critério utilizado na metodologia sugere que a utilização direta da distância euclidiana correspondem à uma medida de dissimilaridade de rosto. Uma vez que esta distância é produzida, tarefas como reconhecimento de rosto, verificação e agrupamento podem ser facilmente implementadas usando técnicas comuns utilizando como *input*, os vetores de características.

O método usa uma rede convolucional profunda treinada para otimizar diretamente o *encoding* em si, em vez de uma camada intermediária de gargalo (como em abordagens anteriores de aprendizado profundo). Para treinar, são utilizados trios de "patches de rosto" correspondentes/não correspondentes gerados usando um novo método de mineração de trios online. A vantagem dessa abordagem é a maior eficiência representacional: o sistema alcança desempenho de reconhecimento de rosto de última geração usando apenas 128 bytes por rosto.

No amplamente utilizado conjunto de dados *Labeled Faces in the Wild* (LFW), o sistema atinge uma nova precisão recorde de 99,63%. No YouTube Faces DB, ele atinge 95,12%. O sistema reduz a taxa de erro em comparação com o melhor resultado publicado em 30% em ambos os conjuntos de dados.

O artigo também introduz o conceito de *encodings* harmônicos e uma perda de trios harmônicos, que descrevem diferentes versões de *encodings* e rosto (produzidos por diferentes redes) que são compatíveis entre si e permitem comparação direta entre si.

O método é baseado em aprender um *encoding* por imagem usando uma rede convolucional profunda. A rede é treinada de forma que as distâncias quadradas L2 no espaço de "encodings" que correspondam diretamente à similaridade do rosto: rostos da mesma pessoa têm pequenas distâncias e rostos de pessoas distintas têm grandes distâncias, naturalmente.

Como saída da rede, a FaceNet treina diretamente sua saída para ser um array compacto de 128 espaços usando uma função de perda baseada em trios com base no LMNN.

Estes "patches" dos trios consistem em miniaturas de rosto correspondentes e uma miniatura de rosto não correspondente e a perda visa separar o par positivo do negativo por uma margem de distância.

A escolha de quais trios usar se mostra muito importante para alcançar um bom desempenho e, inspirado pela aprendizagem curricular, é apresentada uma nova estratégia de mineração de exemplares negativos online que garante uma dificuldade consistentemente crescente de trios à medida que a rede treina.

2.1.4 VGG16

A Very Deep Convolutional Networks for Large-Scale Image Recognition, mais conhecida como VGG, é uma arquitetura de rede neural convolucional que se destacou na competição *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* de 2014. A VGG foi desenvolvida por pesquisadores da *Visual Geometry Group* e da Universidade de Oxford, daí o nome "VGG". A principal contribuição da VGG é a demonstração de que a profundidade da rede (ou seja, o número de camadas na rede) é um componente crítico para o bom desempenho de uma CNN (SIMONYAN; ZISSERMAN, 2014).

A arquitetura da VGG é caracterizada por sua simplicidade, usando apenas camadas convolucionais 3x3 empilhadas profundamente na rede, com camadas de *pooling* máximas para redução de dimensionalidade. A VGG é composta por 16 a 19 camadas e, embora seja uma rede muito profunda, ela tem uma arquitetura muito uniforme e sistemática. A VGG usa um grande número de filtros (até 512) em suas camadas mais profundas.

A VGG foi treinada em um conjunto de dados de milhões de imagens com milhares de categorias de objetos. A rede aprendeu a reconhecer esses objetos com uma precisão relevante, superando muitas outras arquiteturas de CNN na competição ILSVRC, alcançando o segundo lugar na tarefa de classificação de imagem e o primeiro lugar na tarefa de localização de objeto. A VGG conseguiu uma taxa de erro top-5 de 7,3% no conjunto de dados de validação do ImageNet, que foi significativamente melhor do que os resultados alcançados por outras arquiteturas de rede no mesmo período. No entanto, a VGG é computacionalmente intensiva para treinar e requer uma quantidade significativa de memória para armazenar os pesos da rede.

Apesar de sua intensidade computacional, a VGG é amplamente utilizada na comunidade de aprendizado profundo devido à sua excelente capacidade de generalização. A VGG pode ser aplicada a uma variedade de tarefas de visão computacional, incluindo reconhecimento

de objetos, localização de objetos e detecção de objetos. Além disso, os pesos pré-treinados da VGG são frequentemente usados como ponto de partida para treinar redes para novas tarefas, um processo conhecido como transferência de aprendizado.

A VGG também é frequentemente usada como uma base para a extração de características em tarefas de visão computacional. As características extraídas das camadas intermediárias da VGG são conhecidas por serem muito eficazes na representação de informações visuais complexas. Essas características podem ser usadas para uma variedade de tarefas, desde a classificação de imagens até a detecção de objetos e a segmentação semântica.

Embora a VGG seja uma arquitetura poderosa, ela tem suas limitações. Como mencionado anteriormente, a VGG é computacionalmente intensiva e requer uma quantidade significativa de memória. Isso torna a VGG menos ideal para aplicações em tempo real ou em dispositivos com recursos limitados. Além disso, embora a VGG seja profunda, ela não é muito eficiente em termos de parâmetros. A VGG tem um grande número de parâmetros, o que pode levar a um sobreajuste se não for gerenciado corretamente.

A VGG também é conhecida por ser relativamente lenta para treinar e inferir em comparação com outras arquiteturas de CNN. Isso se deve em grande parte ao grande número de parâmetros na rede e à profundidade da rede. No entanto, apesar dessas limitações, a VGG continua sendo uma arquitetura popular e influente na comunidade de aprendizado profundo devido à sua simplicidade e desempenho impressionante em tarefas de visão computacional.

2.1.5 GoogLeNet

A GoogLeNet introduz uma arquitetura de rede neural convolucional profunda que usa uma nova operação chamada "Inception Module". Este módulo permite que a rede aprenda diferentes características em diferentes escalas, tornando a rede mais eficiente e eficaz (SZEGEDY *et al.*, 2015). A arquitetura da GoogLeNet é significativamente mais profunda do que as arquiteturas anteriores, com 22 camadas de profundidade. Isso permite que a rede aprenda representações mais complexas e abstratas dos dados de entrada.

Além disso, a GoogLeNet é projetada para ser eficiente em termos de computação e memória, o que a torna adequada para dispositivos com recursos limitados. A arquitetura é modular, o que significa que os módulos de iniciação podem ser empilhados para criar redes de diferentes profundidades e larguras. Isso permite que a GoogLeNet seja facilmente adaptada para diferentes tarefas e conjuntos de dados.

Em relação complexidade computacional, é utilizada uma outra abordagem dentro da rede. Através da substituição de camadas totalmente conectadas por camadas de convolução média, o que reduz o número de parâmetros e, portanto, a complexidade computacional, a rede consegue obter um desempenho adicional. A redução da complexidade computacional é crucial para tornar a GoogLeNet viável para treinamento em hardware disponível comercialmente.

Esse ajuste na complexidade computacional também permite que a GoogLeNet seja mais eficiente em termos de memória, pois requer menos parâmetros do que as arquiteturas de rede neural convolucional tradicionais. Isso torna a GoogLeNet mais adequada para dispositivos com recursos limitados, como smartphones e tablets.

A GoogLeNet introduz conceito de utilizar um classificador auxiliar para combater o problema do desaparecimento do gradiente em redes profundas. Este classificador é adicionado a uma camada intermediária da rede e é usado apenas durante o treinamento para propagar o gradiente de volta à rede. O classificador auxiliar ajuda a garantir que o sinal do gradiente não se atenuie à medida que é propagado de volta através das camadas da rede durante o treinamento.

O uso de um classificador auxiliar é uma inovação importante que permite que a GoogLeNet seja treinada efetivamente, apesar de sua profundidade. Isso permite que a rede aprenda representações mais complexas e abstratas dos dados de entrada, o que melhora o desempenho da rede na tarefa de classificação.

Em termos de eficiência de memória, a GoogLeNet também é projetada para ser eficiente na adequação do modelo em relação a dispositivos com recursos limitados. A arquitetura da GoogLeNet é modular, o que significa que os módulos de iniciação podem ser empilhados para criar redes de diferentes profundidades e larguras. Isso permite que a GoogLeNet seja facilmente adaptada para diferentes tarefas e conjuntos de dados, como também oferece mais portabilidade para trabalhar desde smartphones e tablets até servidores de alto desempenho.

A GoogLeNet superou outras arquiteturas de rede neural convolucional no desafio de classificação de imagem ImageNet 2014, demonstrando a eficácia de sua arquitetura profunda e módulos de iniciação. A GoogLeNet conseguiu uma taxa de erro top-5 de 6,67% no conjunto de dados de validação do ImageNet, que foi significativamente melhor do que os resultados alcançados por outras arquiteturas de rede na época. Nesse sentido, a GoogLeNet foi capaz de aprender representações mais complexas e abstratas dos dados de entrada, superando demais arquiteturas de rede na tarefa de classificação.

2.1.6 Attention

As redes neurais convolucionais são uma classe de redes neurais profundas que têm demonstrado um desempenho excepcional em tarefas de visão computacional, como classificação de imagens, detecção de objetos e segmentação semântica.

No entanto, as CNNs tradicionais tratam todas as regiões da imagem de entrada igualmente, o que pode não ser ideal para algumas tarefas que requerem uma atenção mais focada em certas regiões. Por exemplo, na detecção de objetos, pode ser mais importante se concentrar nas regiões da imagem onde os objetos estão presentes.

Para resolver este problema, os pesquisadores introduziram o conceito de módulos de atenção para CNNs. Um módulo de atenção é uma componente adicional que pode ser incorporada em uma CNN para permitir que ela se concentre mais nas regiões relevantes da imagem.

O módulo de atenção funciona aprendendo a atribuir diferentes pesos às diferentes regiões da imagem. As regiões que são consideradas mais relevantes para a tarefa em questão recebem pesos mais altos, o que significa que a CNN prestará mais atenção a elas.

Existem várias maneiras de implementar um módulo de atenção. Uma abordagem comum é usar uma rede neural totalmente conectada (FCN) para aprender os pesos de atenção. A FCN recebe a imagem de entrada e produz uma mapa de atenção, que é então usado para ponderar a imagem de entrada.

Outra abordagem é usar uma rede neural recorrente (RNN) para aprender os pesos de atenção. A RNN é capaz de processar a imagem de entrada sequencialmente, o que pode ser útil para tarefas que requerem uma compreensão temporal da imagem, como a análise de vídeo.

Os módulos de atenção têm demonstrado melhorar o desempenho das CNNs em várias tarefas. Por exemplo, eles têm sido usados para melhorar a precisão da classificação de imagens, permitindo que a CNN se concentre nas partes da imagem que são mais relevantes para a classe de interesse.

Além disso, os módulos de atenção também têm sido usados para melhorar a precisão da detecção de objetos, permitindo que a CNN se concentre nas regiões da imagem onde os objetos estão presentes.

Os módulos de atenção também podem ser usados para melhorar a interpretabilidade das CNNs. Ao visualizar os mapas de atenção, os pesquisadores podem ganhar uma melhor compreensão de quais regiões da imagem a CNN está se concentrando, o que pode ajudar a

explicar por que a CNN está fazendo certas previsões.

2.2 Processamento de imagens - Segmentação

A segmentação de imagens é um componente crítico e fundamental do processamento de imagens. Ela tem como objetivo dividir uma imagem em regiões ou objetos distintos, cada um dos quais correspondendo a algo de interesse (GONZALEZ; WOODS, 2017). A segmentação de imagens é uma tarefa desafiadora, pois envolve a identificação de padrões que podem ser altamente variáveis e que estão sujeitos a ruído e outras distorções.

Há vários algoritmos e abordagens comumente usados na segmentação de imagens. Um dos métodos mais antigos é a segmentação por limiar, em que os pixels são agrupados com base em seu valor de intensidade. Outros métodos incluem a segmentação baseada em região, que divide a imagem em regiões conectadas com base em critérios de similaridade, e a segmentação baseada em bordas, que procura por descontinuidades na imagem (GONZALEZ; WOODS, 2017).

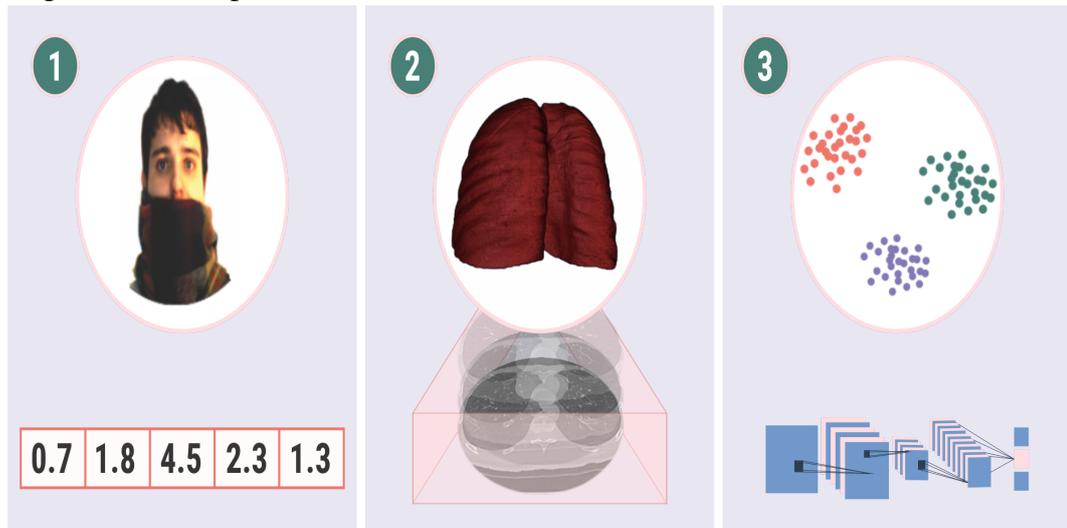
O método de segmentação a ser escolhido depende fortemente da natureza do problema. Alguns problemas podem ser adequadamente resolvidos com métodos simples de limiar, enquanto outros podem exigir métodos mais sofisticados, como a segmentação baseada em grafos, que representa uma imagem como um grafo e então particiona esse grafo para formar segmentos (SHI; MALIK, 2000).

Também é importante destacar que a segmentação de imagens é um campo de pesquisa ativo, com muitas abordagens novas e promissoras sendo propostas. Por exemplo, as técnicas de aprendizado de máquina, e em particular as redes neurais convolucionais, mostraram-se muito eficazes na segmentação de imagens, particularmente para problemas complexos, como a segmentação semântica em imagens de alta resolução (LONG J., 2015).

3 METODOLOGIA

A metodologia deste estudo está focada primariamente em aplicar e avaliar o FoL em diferentes cenários do ponto de vista de otimização, com foco em processamento de imagens e modelos de aprendizado profundo, com e sem pré-treinamento, conforme ilustrado na Figura 5. Dessa forma, o estudo comporta uma avaliação para arrays, imagens e deep-learning: 1D, 2D e multi-dimensão, respectivamente.

Figura 5 – FoL aplicado a dados



Fonte: Elaborada pelo autor.

Para investigar o impacto do FoL e avaliá-lo na acurácia, tempo, na análise de agrupamento e no tamanho do arquivo contendo as *features* salvas provenientes dos modelos na tentativa de avaliar o grau de compressão atingido. Assim, a metodologia está dividida em 5 partes:

1. *Datasets*
2. Introdução ao método FoL.
3. Avaliação do FoL em *array*.
4. Avaliação do FoL em imagens.
5. Avaliação do FoL em multi-dimensão.

3.1 Datasets

Para avaliar o FoL em diferentes cenários, foram utilizados *datasets* que apresentam características distintas, abordando várias nuances do processamento de imagens e aprendizado de máquina. Assim, foram utilizados oito *datasets*, quatro focados em características faciais e 2 focados em imagens de TC de pulmão e mais 2 para benchmark.

Para avaliação facial, os seguintes conjuntos de dados foram empregados: LFW, AR, Extended Yale B e CelebA. LFW (*Labeled Faces in the Wild*) contém 13.233 imagens de 5.749 pessoas, coletadas da internet, sendo um conjunto de dados desafiador devido à variação de pose, iluminação e expressões faciais. Esses datasets distintos são relevantes para o estudo em questão, pois apresentam diferentes características na imagem, como oclusão, artefatos, ruídos etc. AR oferece um conjunto de dados de 3.120 imagens de 126 indivíduos, com diferentes condições de iluminação, expressões faciais e oclusões por óculos e lenços. *Extended Yale B* (EYaleB) possui 16.128 imagens de 28 indivíduos em 9 posições diferentes e 64 variações de iluminação. CelebA é um conjunto de dados com 202.599 imagens de rosto de 10.177 celebridades, ideal para reconhecimento facial e treinamento de modelos de aprendizado profundo. Uma visão geral dos *datasets* é ilustrado pela Figura 6

Para imagens, dois *datasets* gerados a partir de tomografias computadorizadas foram utilizados:

- LUNA16 (LUNG Nodule Analysis 2016): Este conjunto de dados consiste em 888 varreduras de TC de alta resolução coletadas na fase inspiratória de 888 pacientes. Foi projetado para auxiliar na detecção de nódulos pulmonares para diagnóstico precoce do câncer de pulmão.
- LAPISCO: Este é um conjunto de dados brasileiro desenvolvido pelo Laboratório de Processamento de Imagens, Sinais e Computação Aplicada (LAPISCO).

Um processo experimental foi realizado no banco de dados usando as varreduras de TC do desafio LUNA16. O conjunto de dados do LUNA16 oferece uma grande quantidade de imagens de alta qualidade com configurações de aquisição padrão, resultando em um melhor controle de características como brilho e contraste de todas as varreduras de TC.

Embora o processo de segmentação indique bons resultados no conjunto de dados LUNA16, no momento presente deste trabalho, este banco de dados não possui o *Ground Truth* (GT) dos pulmões segmentados para aplicar as métricas de avaliação. Entretanto, sua presença torna-se essencial para treino do FoL para esse tipo de aplicação.

Figura 6 – Figura com exemplos de imagens faciais de cada conjunto de dados usados: AR, Extended YaleB, LFW e CelebA, respectivamente.



Fonte: Elaborada pelo autor.

Para uma comparação de resultados, este trabalho usa o banco de dados LAPISCO-LUNG, que pode ser acessado em <https://lapisco.ifce.edu.br/>. Este conjunto de dados foi composto por 40 imagens de tomografia computadorizada da região do tórax, contendo o GT para cada *slice* do exame de TC.

A abordagem proposta é avaliada a partir do conjunto de imagens obtido em parceria com o Hospital Walter Cantídio da Universidade Federal do Ceará. Os estudos neste conjunto de dados foram autorizados pelo Comitê de Ética em Pesquisa da UFC – COMEPE, protocolo 35/06.

Este conjunto de imagens foi previamente avaliado para classificações de doenças (ALVES *et al.*, 2018), (REBOUÇAS *et al.*, 2017c), (PEIXOTO *et al.*, 2018), (RAMALHO *et al.*, 2016), reconhecimento automático do pulmão (REBOUÇAS *et al.*, 2017a), (??), (MEDEIROS *et al.*,) e detecção de doenças pulmonares (RAMALHO *et al.*, 2014). Todas as amostras foram coletadas dos equipamentos Toshiba Aquilion GE Medical System LightSpeed16 e Philips Brilliance 10. O conjunto é formado por 72 imagens pulmonares, 512 x 512 pixels com 16 bits cada. *Ground Truth* da segmentação foi produzida por uma análise manual de um especialista

experiente.

3.2 *Floor of Log*

O método *Floor of Log* é uma operação matemática que integra a função chão e o logaritmo de um valor ajustado por um *bias*, empregando uma base específica que deve ser adaptada para cada aplicação. Esse método é definido pela Equação 3.1, onde x é o valor de entrada, *bias* é uma "constante" opcional adicionada a x , e a base do logaritmo é indicado por b . A função *floor* arredonda o valor para baixo, para o inteiro mais próximo, discretizando o resultado. Essa função pode ser aplicada a vários tipos de dados e tem o potencial de reduzir a complexidade dos dados enquanto retém informações importantes.

$$f(x) = \lfloor \log_b(x + \text{bias}) \rfloor, \quad (3.1)$$

Entretanto, nota-se que são necessários a implementação de algumas restrições importantes para garantir sua validade matemática, como sugerido pela equação 3.2, que determina o domínio do método. Nesse caso, a variável x não pode ser igual a sinal inverso do *bias*, pois isso resultaria em uma situação em que o argumento do logaritmo seria igual a zero, causando problemas matemáticos, como a indefinição do logaritmo.

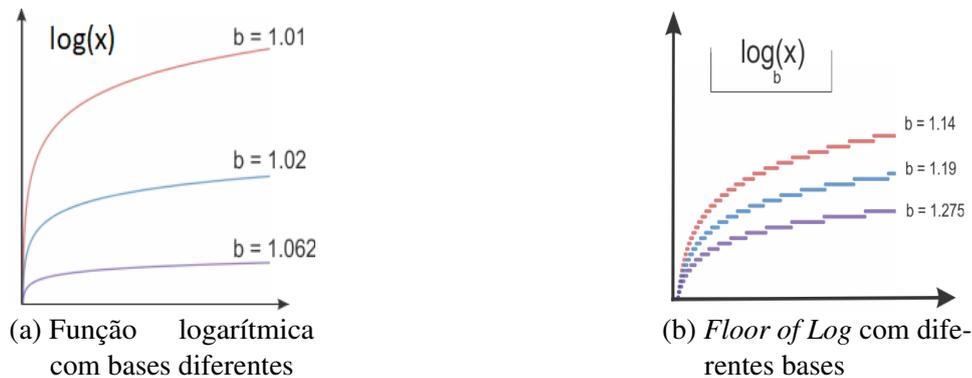
$$f(x) : x \neq -\text{bias}, \quad b > 0, \quad b \neq 1, \quad \text{bias} + x > 0 \quad (3.2)$$

Além disso, a base do logaritmo, pode ser um número real maior que zero e diferente de 1. A restrição de b ser maior que zero assegura que a base seja positiva, evitando situações de logaritmos com bases negativas, que são complexas e não se aplicam a este contexto. A exclusão de b igual a 1 é importante, pois o logaritmo na base 1 é indefinido para qualquer valor de x diferente de 1, uma vez que a função logarítmica se torna constante e não possui uma solução única.

Na Figura 7, podemos observar o efeito da função chão aplicada à função logarítmica com diferentes valores de bases. Nesta figura, 7a apresenta diversas funções logarítmicas com bases distintas, enquanto 7b ilustra a aplicação da função chão ao logaritmo com diferentes valores de bases. Essa representação gráfica nos permite compreender a relação entre a função chão e a função logarítmica, bem como o impacto de variar a base da função logarítmica. Apesar

de trivial, a Figura 7b dá indícios de um comportamento de truncamento ou mesmo clusterização, conceito que será utilizado amplamente durante esse trabalho.

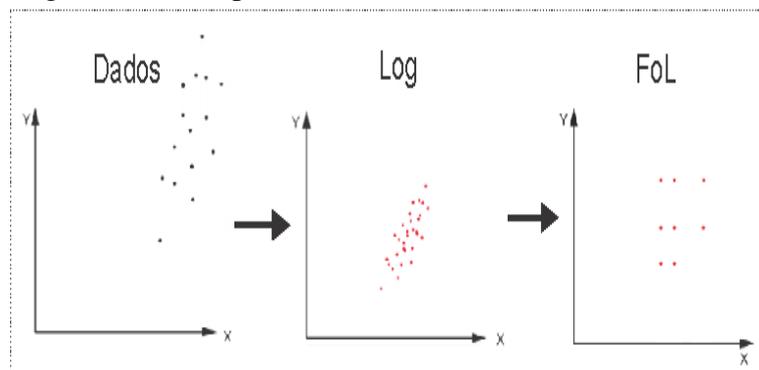
Figura 7 – Efeito da função chão aplicada na função logarítmica com diferentes valores de bases. Em (a), algumas funções logarítmicas com bases diferentes. Em (b), a função chão aplicada ao logaritmo com diferentes valores de bases



Fonte: Elaborada pelo autor.

Do ponto de vista de um conjunto de dados, o FoL promove alguns efeitos de clusterização como ilustrado na Figura 8. Como tem por definição uma função logarítmica, essa transformação não linear é capaz de aproximar valores que estão próximos uns dos outros. Em seguida, uma função chão é aplicada à função logarítmica, de modo que apenas o limite inferior dessa função seja preservado, ou seja, apenas a sua parte inteira. É notável, portanto, que um aspecto a ser extremamente levado em consideração é a escolha da base, pois ela quem "guiará" a força do agrupamento.

Figura 8 – FoL aplicado a dados

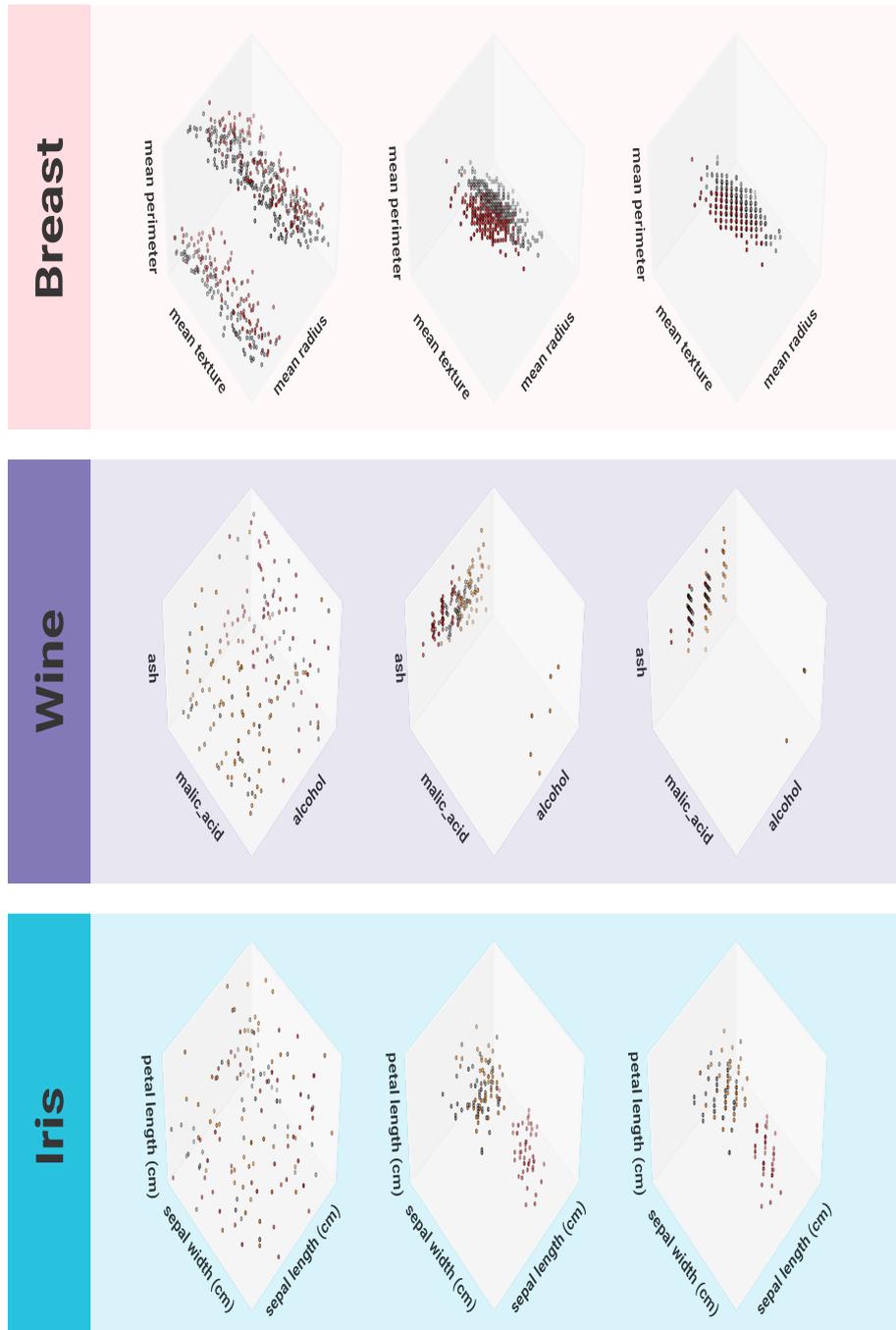


Fonte: Elaborada pelo autor.

Na aplicação direta em *datasets* convencionais, poderemos ver como a compressão se

comportaria quando aplicada às *features*. A Figura 9 ilustra o comportamento do FoL aplicado a 3 *datasets* com diferentes bases. Podemos observar que além das *features* estarem se aproximando entre si, parte delas acabam se sobrepondo. Sendo assim, seria interessante avaliar se, nesse caso, o *dataset* pudesse ter uma redução de dados repetidos, depois de transformados.

Figura 9 – FoL aplicado no *datasets* *Breast*, *Wine* e *Iris* em diferentes bases.



Fonte: Elaborada pelo autor.

Ao aplicar a função FoL em diferentes ambientes a seguir, pretendemos obter uma

compreensão abrangente de sua influência em vários aspectos do processamento de imagens e modelos de aprendizado profundo. Isso ajudará a identificar os casos de uso mais adequados para a função FoL. Além disso, a exploração de FoL em várias configurações contribui para o desenvolvimento de técnicas de processamento de imagem e aprendizado profundo mais eficientes e precisas, melhorando o desempenho de sistemas de visão computacional e aprendizado de máquina em uma ampla gama de aplicativos.

3.3 Análise da Complexidade Assintótica

Nesta seção, apresentamos a análise da complexidade temporal do algoritmo *Floor of Log*, ou seja, do cálculo de $\lfloor \log_{\text{base}}(x + \text{bias}) \rfloor$. Para isso, analisaremos o número de operações necessárias para calcular a função em questão. Essa análise nos permite entender o custo computacional do método proposto e, conseqüentemente, a justificativa imediata do porque da independência da dimensionalidade.

Considere que temos n elementos (por exemplo, pixels em uma imagem ou valores em um vetor) para os quais queremos aplicar a função *Floor of Log*. Para cada elemento x_i , temos as seguintes etapas:

1. Calcular a soma: $x_i + \text{bias}$
2. Calcular o logaritmo: $\log_{\text{base}}(x_i + \text{bias})$
3. Calcular o chão: $\lfloor \log_{\text{base}}(x_i + \text{bias}) \rfloor$

Em cada uma dessas etapas, realizamos uma operação aritmética simples (adição, logaritmo e chão) para cada elemento x_i . Portanto, a complexidade temporal para cada elemento é $O(1)$, ou seja, constante. Como temos n elementos, a complexidade temporal total para todos os elementos será $O(n)$.

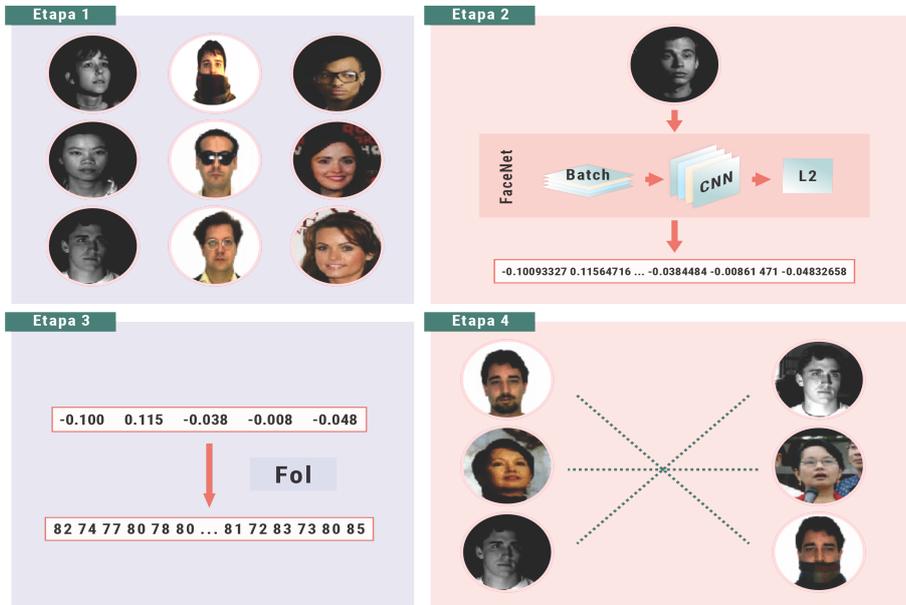
Assim, podemos concluir que a complexidade temporal do algoritmo *Floor of Log* é linear, ou seja, $O(n)$, onde n é o número de elementos aos quais a função é aplicada. Sendo assim, independente da disposição do valor de entrada, perante a sua dimensionalidade, ao aplicar o FoL, sua dimensionalidade em questão será mantida.

3.4 Avaliação do FoL em *array*

Nesta seção, discutiremos a metodologia abordada para avaliar o FoL em um modelo pré-treinado, nesse caso, a FaceNet. As imagens utilizadas, para essa aplicação, são todas que

compõem os *datasets* de faces apresentados na Seção 3.1. O processo de treinamento dos parâmetros do método proposto é discutido na subseção 3.4.1. A Figura 10 ilustra a visão geral da metodologia adotada, incluindo a aquisição de dados de cada conjunto de dados e extração das características usando o FaceNet. A terceira etapa foca na compressão dessas *features*. Por fim, o método é avaliado usando métricas de correspondência facial e a comparação do tamanho dos dados antes e depois da compactação.

Figura 10 – Visão geral da metodologia adotada. Primeiro, todos os conjuntos de dados faciais são carregados, seguidos pela extração de features pelo FaceNet. Na terceira etapa, a abordagem proposta é aplicada às características, reduzindo seu tamanho mantendo a qualidade. A quarta etapa apresenta a etapa de avaliação, que, neste caso, é baseada na correspondência de faces.



Fonte: Elaborada pelo autor.

As características extraídas obterão valores de estrutura semelhantes após a aplicação da função chão, que removerá a parte fracionária dos dados. A equação 3.3 abaixo representa a função principal do método de compressão, onde b é a base logarítmica da função \log . Como as características extraídas são descritas em casas decimais, impactando na função \log , o *bias* é usado para corrigir esse valor.

$$f(x) = \lfloor \log_b(\text{Features} + \text{Bias}) \rfloor \quad (3.3)$$

3.4.1 Processo de treinamento

Uma vez que cada classe corresponde a face da amostra, o treino para encontrar a base do FoL reduz-se a testar diferentes bases até que o conjunto de treinamento possua o maior *score* dentre as demais bases. Assim, como o FoL é aplicado diretamente nas features, apenas ajustá-lo quanto ao *bias* e tratar a aplicação como um problema comum de classificação facial seria o suficiente. O Algoritmo 1 sugere uma abordagem de força bruta com um classificador para avaliar a qualidade da base em cada iteração.

Algorithm 1 Algoritmo proposto para encontrar uma referência para base logarítmica

Input: *data*, *step*, *nIterations*, *classifier*, *kFold*, *bias*
lastScore \leftarrow 0
bestBase \leftarrow *base* \leftarrow 1.0001
for *i* \leftarrow 0 to *nIterations* **do**
 clusterizedValues \leftarrow *floor*(*log*(*data*, *base*, *bias*))
 x, *y* \leftarrow *removeDuplicates*(*clusterizedValues*)
 stats \leftarrow *crossValidation*(*classifier*, *x*, *y*, *kFold*)
 meanScore \leftarrow *mean*(*stats.accuracy*)

 if *meanScore* \geq *lastScore* **then**
 lastScore \leftarrow *meanScore*
 bestBase \leftarrow *base*
 end if
 base \leftarrow *base* + *step*
end for

Finalmente, o algoritmo Support Vector Machine (SVM) é empregado como classificador, realizando validação cruzada e criando um hiperplano que separa as características mais relevantes de cada classe. Este método é robusto mesmo na presença de artefatos ou ruídos, sendo uma ferramenta poderosa no aprendizado de máquina.

3.5 Avaliação do FoL em imagens

Nesta seção, exploramos a aplicação do FoL em imagens, especificamente na segmentação pulmonar utilizando imagens médicas oriundas de tomografias computadorizadas. Vale ressaltar que, neste contexto, o valor de *bias* pode representar um dado ou filtro aplicado inicialmente à imagem.

No entanto, mesmo após a aplicação da transformada logarítmica, alguns artefatos ou

ruídos na imagem podem prejudicar o processo de agrupamento. Para contornar esse problema, a utilização de filtros ou pré-processamento pode auxiliar no agrupamento local de pixels e atenuar a homogeneização das regiões de interesse em relação às áreas externas.

Neste trabalho, optamos por adicionar o *Local Binary Pattern* (LBP) à imagem original antes de aplicar o FoL. A combinação da imagem original com a imagem LBP fornece uma alteração benéfica quanto aos ruídos presentes (ou pequenas variações) a imagem. A imagem resultante é mostrada pela Equação 3.4, onde 'O' representa a imagem original, 'L' a imagem resultante do LBP e 'I' a saída da operação da soma pixel-a-pixel das posições (i,j). Em seguida, mantemos a aplicação do método sobre essa imagem resultante, como mostrado na Equação 3.5. A Figura 11 ilustra o comportamento do processo de clusterização.

$$I_{i,j} = O_{i,j} + L_{i,j}, \quad bias = L \quad (3.4)$$

$$R_{i,j} = \lfloor \log_b(I_{i,j}) \rfloor \quad (3.5)$$

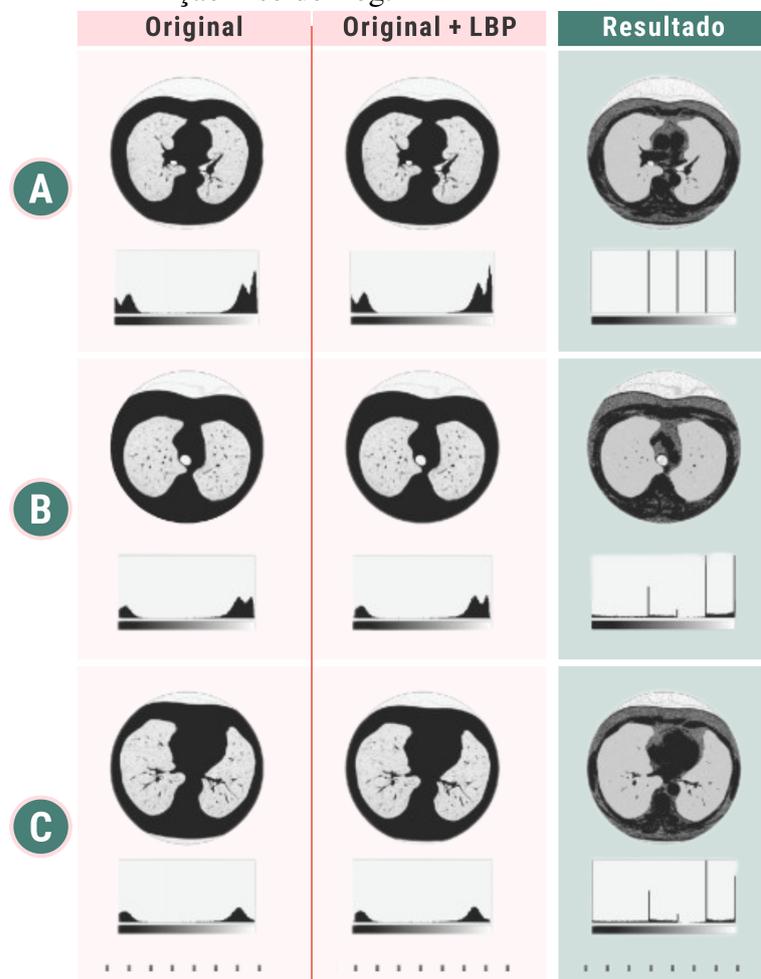
3.5.1 Segmentação

Após a análise do treinamento, a base logarítmica será utilizada na equação 3.1 e assim todo o conjunto de dados será agrupado. Observe que cada cluster oferece um valor único que representa uma parcela dos dados e, nesse sentido, extrairemos das imagens tomográficas o valor correspondente à região agrupada do pulmão. Mesmo assim, algumas regiões permanecem em foco, já que em outras regiões também pode ocorrer acúmulo de gases, como o intestino.

Outro fato conhecido por outros métodos de segmentação pulmonar é a presença do hilo. Essa estrutura geralmente está presente nas imagens segmentadas porque está diretamente conectada ao pulmão e também aerada. Para contornar esse problema, este trabalho utilizou técnicas morfológicas, especificamente a erosão, em que pode ser utilizada para descolar objetos e, portanto, separar o hilo da estrutura pulmonar. As regiões dos vasos sanguíneos e vias aéreas podem ser complementadas com funções simples de abertura e fechamento, conforme ilustrado na Figura 12.

Finalmente, uma vez que os objetos ainda estão separados por clusters, podemos segmentar o pulmão em uma única operação, pois ele, em um espaço tridimensional, tem a maior

Figura 11 – Etapas do processo de clusterização. A, B e C apresenta alguns exemplos do método proposto. Após a imagem de entrada, esta foi combinada com LBP image e depois agrupados usando o algoritmo *Floor of Log*. A coluna O resultado apresenta a imagem normalizada após a operação Piso do Log.



Fonte: Elaborada pelo autor.

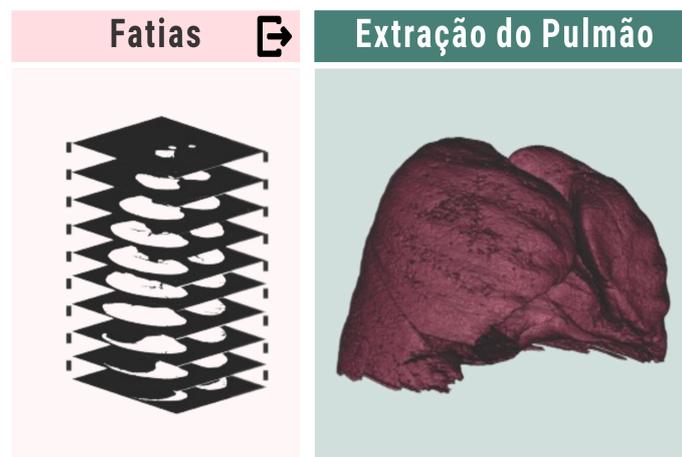
Figura 12 – Operação Flood-Fill aplicada na imagem
1. O resultado foi apresentado pelo ponto
2



Fonte: Elaborada pelo autor.

volume. Portanto, basta que os dois objetos com maior volume (pulmão esquerdo e direito) no espaço tridimensional são extraídos para ter o pulmão segmentado, finalizando o processo. Assim, é necessário que cada uma das imagens processadas seja concatenada em sequência, a fim de analisar o imagem inteira em um espaço tridimensional, conforme indicado pela Figura 13.

Figura 13 – A representação tridimensional da região pulmonar. Depois o processo de cada fatia, o algoritmo extrai todo o pulmão na terceira dimensão usando os 2 volumes maiores.



Fonte: Elaborada pelo autor.

3.5.2 Métricas de avaliação

Para realizar a comparação, Falso Negativo (FN), Verdadeiro Positivo (VP), Falso Positivo (FP) e Verdadeiro Negativo (VN) foram extraídos da matriz de confusão da imagem a partir da abordagem proposta e da segmentação especializada. Cada um desses valores calculados é utilizado nas métricas de avaliação, apresentadas a seguir:

A proporção dos erros totais e a possibilidade de segmentação correta foi avaliada pela Acurácia (Acc), fornecendo um índice qualitativo de como, em termos gerais, o resultado da segmentação foi apresentado. Sua fórmula é dada por:

$$Acc = \frac{VN + VP}{VN + FP + VP + FN} \quad (3.6)$$

Outro índice qualitativo é apresentado pela Sensibilidade (Se), cuja métrica reside na capacidade de prever corretamente uma região em comparação com um conjunto de regiões que realmente pertencem à região pulmonar, apresentado por:

$$Se = \frac{VP}{VP + FN} \quad (3.7)$$

Outra métrica estatística é o Coeficiente de Correlação de Matthews (MCC). Neste caso, o resultado qualitativo entre a abordagem proposta e a segmentação especializada também é analisado, mesmo com escalas diferentes dessas imagens. A correlação MCC é calculada conforme apresentado abaixo:

$$MCC = \frac{(VN \cdot VP - FN \cdot FP)}{\sqrt{(FP + VP)(FN + VP)(FP + VN)(FN + VN)}}, \quad (3.8)$$

Para a comparação de similaridade, a distância de Hausdorff ((HAUSDORFF, 1914)) foi incluída para análise. Usando o espaço de Hausdorff, a distância euclidiana máxima sujeita à região segmentada pela abordagem proposta em relação ao GT e a distância euclidiana máxima sujeita ao GT em relação à região segmentada é analisada. Pode-se ver que se a região segmentada de ambas as imagens for semelhante, essa distância tende a ser igual. A fórmula HD é apresentada por:

$$HD = \max \{A, B\}, \quad (3.9a)$$

Onde

$$A = \sup_{x \in R_a} \inf_{y \in R_{gt}} d(x, y) \quad B = \sup_{y \in R_{gt}} \inf_{x \in R_a} d(x, y) \quad (3.9b)$$

O coeficiente Dice (DC) (DICE, 1945), também é uma métrica utilizada para verificar a similaridade entre regiões segmentadas. Este método utiliza uma sobreposição de imagens, apresentada como:

$$Dice = 2 \cdot \frac{|S_a \cap S_{gt}|}{|S_a + S_{gt}|}, \quad (3.10)$$

onde S_a representa a segmentação algorítmica, ou seja, a região segmentada pela abordagem proposta, e S_{gt} é a região selecionada pelo especialista. Essas regiões sobrepostas permitem a extração de valores binários após a comparação de cada pixel em ambas as regiões. Pode-se ver que se o valor 1 corresponder à sobreposição perfeita em cada pixel, a soma da correspondência total corresponde à maior similaridade com a segmentação especializada (QIAN *et al.*, 2013).

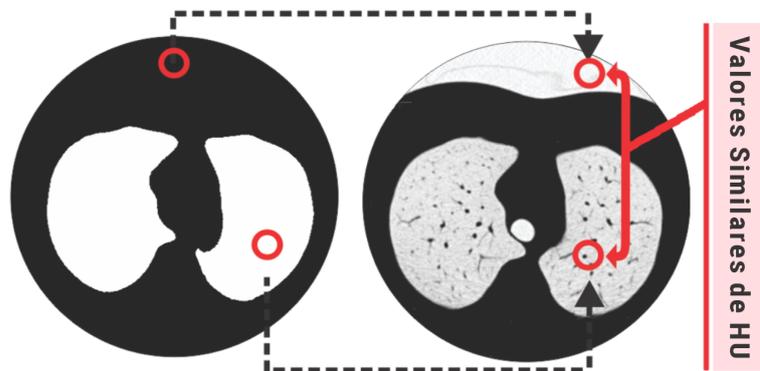
3.5.3 *Treinamento da base logarítmica*

Utilizando um conjunto de imagens segmentadas por um médico especialista, podemos empregar técnicas supervisionadas de inteligência artificial para analisar diferentes valores da base logarítmica e escolher o mais adequado. Essa abordagem normalmente requer uma função de custo e a simulação do gradiente descendente. A função custo, nesse caso, envolve a classificação das intensidades de pixel do objeto de interesse e, como a tomografia computadorizada utiliza unidades Hounsfield para representar valores dos tecidos humanos, é possível separar esses valores para evidenciar a diferença entre regiões pulmonares e não pulmonares, tratando o problema como uma classificação binária. Em suma, precisamos escolher valores para a base que separem os pixels relacionados ao pulmão dos pixels que são divergentes a ele.

Uma abordagem clássica consiste em aplicar a máscara criada pelo especialista diretamente na imagem original, usando os valores associados à máscara como classe pulmonar e ao fundo como classe não pulmonar, conforme ilustrado na Figura 14.

Essa abordagem pode gerar problemas de separação nos algoritmos de classificação devido à similaridade entre regiões internas do pulmão e o entorno do paciente. Para solucionar isso, este trabalho utiliza apenas o contorno da máscara como fundo, relacionando a classe não

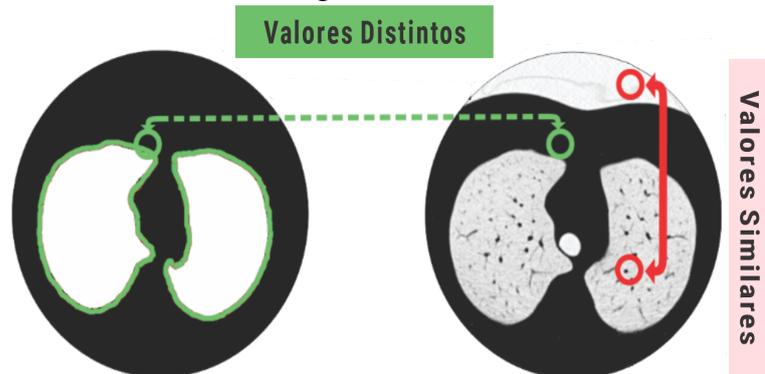
Figura 14 – Problemas na criação do dataset utilizado no processo de training. O entorno e o ROI podem ter os mesmos valores.



Fonte: Elaborada pelo autor.

pulmonar ao parênquima pulmonar e fornecendo valores distintos para análise no treinamento, ilustrado pela Figura 15.

Figura 15 – O contorno verde na imagem do meio apresenta a classe negativa utilizada nesse trabalho.



Fonte: Elaborada pelo autor.

A etapa de treinamento busca minimizar a função de custo iterativamente, atualizando a base logarítmica e evitando clusters desnecessários. O algoritmo que define o processo de treinamento pode ser demonstrado de acordo com o Algoritmo 1.

3.6 Avaliação em modelos gerais

Para incorporar a função FoL nos modelos de aprendizado profundo, primeiro inserimos uma camada personalizada antes ao *flatten* das *features* da rede neural convolucional e antes das camadas totalmente conectadas. Essa camada personalizada calcula a função FoL,

aplicando $\text{floor}(\log(x+\text{bias}, \text{base}))$ antes do *flatten*.

Como a parte do chão da função FoL não possui derivada, usamos a técnica *Straight-Through Estimator* (STE) para ignorá-la durante o *backpropagation*. Isso garante que o fluxo de gradiente permaneça ininterrupto, permitindo o treinamento eficiente dos modelos. Essa modificação pode ser avaliada na Equação 3.11.

$$\text{FoL}_{\text{STE}}(x) = \begin{cases} \text{FoL}(x + \text{bias}) & \text{Durante o } forward \text{ pass} \\ \log_{\text{base}}(x + \text{bias}) & \text{Durante o } backward \text{ pass} \end{cases} \quad (3.11)$$

Para otimizar as variáveis base e *bias* nos modelos de aprendizado profundo modificados, empregamos o algoritmo de otimização Adadelta. Adadelta é um método de taxa de aprendizado adaptável que é adequado para lidar com dados esparsos e grandes conjuntos de dados. Ao usar o Adadelta, pretendemos obter melhor convergência e desempenho do modelo ao incorporar a camada FoL. Nesse sentido, a reorganização do Adadelta apresentada pela Equação 3.12 e alterada conforme o fluxo abaixo:

1. Cálculo do gradiente referente a x: $g_{\text{normal}_x} = \frac{d(\lfloor \log_b(x+\text{Bias}) \rfloor)}{dx}$
2. Cálculo do gradiente referente a x: $g_{\text{ste}_x} = \frac{d(\log(x+\text{bias},\text{base}))}{dx}$
3. Cálculo do gradiente referente a bias: $g_{\text{ste}_{\text{bias}}} = \frac{d(\log(x+\text{bias},\text{base}))}{dbias}$
4. Cálculo do gradiente referente a base: $g_{\text{ste}_{\text{base}}} = \frac{d(\log(x+\text{bias},\text{base}))}{dbase}$

$$\theta_{t+1} = \theta_t - \frac{\text{RMS}(\Delta\theta_t)}{\text{RMS}(g_{\text{ste}_t})} \cdot g_{\text{ste}_t} \quad (3.12)$$

3.6.1 Conjunto de dados

Os modelos de aprendizado profundo modificados são treinados e testados em dois conjuntos de dados de referência, CIFAR-10 e CIFAR-100. Esses conjuntos de dados são amplamente utilizados para avaliar algoritmos de classificação de imagens e fornecem uma plataforma adequada para avaliar o impacto da camada FoL no desempenho dos modelos.

3.6.2 Métricas de avaliação

Para avaliar o impacto da camada FoL nos modelos de aprendizado profundo, usamos três métricas de avaliação principais: acurácia, análise de agrupamento não supervisionada e tamanho dos dados comprimidos.

A acurácia fornece uma medida direta do desempenho do modelo na classificação de imagens, enquanto a análise de clustering não supervisionada ajuda a avaliar a qualidade das *features* aprendidas em termos de sua capacidade de formar clusters distintos. Para análise de agrupamento, empregamos os escores de Calinski-Harabaz(CH), Davies-Bouldin(DB) e Silhouette(S).

O índice CH é uma razão entre a dispersão entre clusters e a dispersão dentro do cluster. Valores mais altos do índice CH indicam melhor qualidade de agrupamento. A fórmula para o índice de Calinski-Harabasz é dada como:

$$CH = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1} \quad (3.13)$$

onde B_k é a matriz de dispersão entre clusters, W_k é a matriz de dispersão dentro do cluster, k é o número de clusters, N é o número de pontos de dados e Tr denota o traço de uma matriz.

O índice DB avalia a qualidade do agrupamento com base na medida de similaridade média de cada cluster com seu cluster mais semelhante. Valores mais baixos do índice de banco de dados significam melhor agrupamento. A fórmula para o índice de Davies-Bouldin é dada como:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left(\frac{S_i + S_j}{d_{ij}} \right) \quad (3.14)$$

onde k é o número de clusters, S_i e S_j são as distâncias médias dos pontos de dados dentro dos clusters i e j , respectivamente, e d_{ij} é a distância entre os centróides dos clusters i e j .

A métrica *Silhouette* mede a coesão e a separação de clusters comparando a distância média intra-cluster com a distância média para o cluster vizinho mais próximo. As pontuações de silhueta variam de -1 a 1, onde valores mais altos indicam melhor qualidade de agrupamento. A fórmula para a pontuação do Silhouette para um único ponto de dados é dada como:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3.15)$$

onde $a(i)$ é a distância média do ponto de dados i para todos os outros pontos de dados no mesmo cluster, e $b(i)$ é a distância média mínima do ponto de dados i para todos os pontos de dados em qualquer outro cluster, excluindo o cluster em que i está.

A pontuação geral do *Silhouette* é então calculada como a média das pontuações do *Silhouette* para todos os pontos de dados.

O tamanho do arquivo das *features* salvas serve como um canal de avaliação para a complexidade do modelo e as *features*. Ao comparar os tamanhos de arquivo com e sem a camada FoL, podemos avaliar o efeito da função FoL na eficiência do modelo e nas demandas computacionais.

4 RESULTADOS

Nesta seção, apresentaremos as descobertas e os resultados das três avaliações distintas da técnica *Floor of Log* em diferentes áreas de tecnologia. O objetivo é avaliar o impacto e a eficácia do FoL em vários contextos e compará-lo com métodos tradicionais ou alternativos. Ao estruturar a seção de acordo com as vertentes das Sessões 3.4, 3.5 e 3.6, podemos fornecer uma análise abrangente dos resultados e extrair *insights* significativos. A Figura 16 ilustra um *overview* sobre a metodologia aplicada para gerar os resultados.

4.0.1 Configurações gerais dos parâmetros utilizados

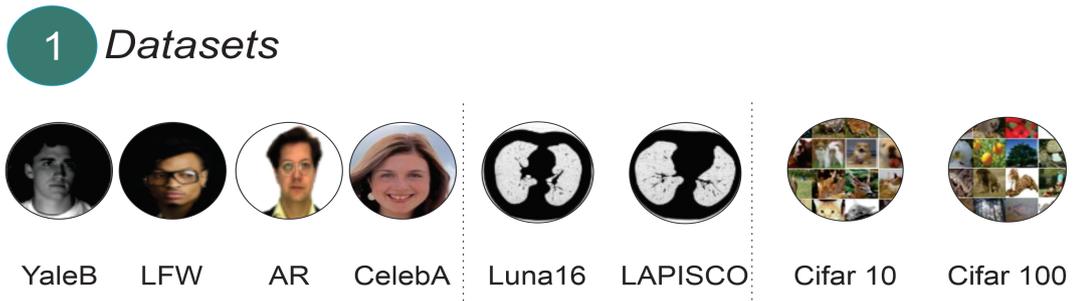
Nesta seção as configurações gerais para a execução do método estão descritas. O objetivo é observar, de uma forma geral, que os parâmetros do FoL podem ser entendidos como um problema similar ao de regressão, compondo características que simulam um gradiente, baseando-se em uma função custo.

Para o processamento de imagens, o ponto de aplicação para essas imagens foi a imagem resultante, e o otimizador utilizado foi a força bruta. A taxa de aprendizado foi definida em 10^{-4} , e o modelo foi treinado por 100 épocas. A validação cruzada foi escolhida com um k igual a 10.

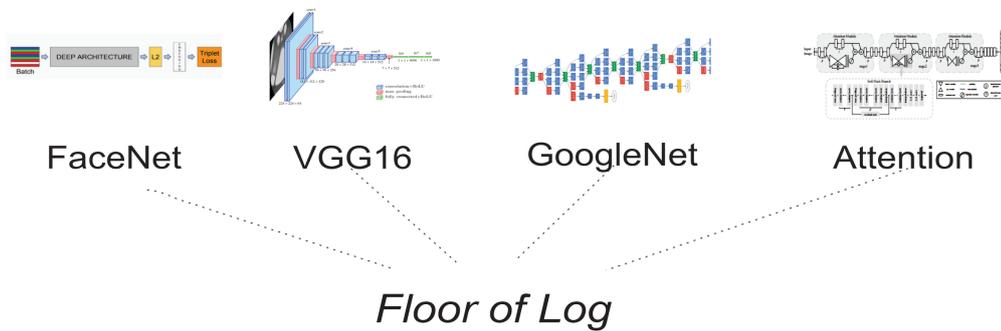
Para o modelo Facenet, os conjuntos de dados AR, LFW, EYaleB e CelebA foram utilizados. Esses conjuntos de dados contêm características faciais e são usados para reconhecimento facial e treinamento de modelos de aprendizado profundo. O ponto de aplicação para esses conjuntos de dados foi a saída do modelo, e o otimizador utilizado também foi a força bruta. A taxa de aprendizado foi definida em 10^{-4} , e o modelo foi treinado por 200 épocas. A validação cruzada foi escolhida com um k igual a 10.

Para as CNNs da terceira aplicação, os conjuntos de dados utilizados foram Cifar10 e Cifar100. Esses conjuntos de dados são amplamente utilizados para avaliar algoritmos de classificação de imagens. O ponto de aplicação para esses conjuntos de dados foi dentro do modelo, e o otimizador utilizado foi AdaDelta. A taxa de aprendizado, inicial, foi definida em 10^{-4} , e o modelo foi treinado por 120 épocas. O tamanho do lote foi definido em 64.

Figura 16 – Overview do processo proposto para gerar os resultados. Inicialmente, para cada problema os datasets são coletados e posteriormente direcionados para cada aplicação. Em seguida, o FoL é aplicado em cada saída dos métodos. Finalmente, para cada aplicação os resultados específicos de cada uma serão avaliados.



2 Aplicação do FoL em múltiplas abordagens



3 Métricas de Avaliação

- Acurácia
- Métricas qualitativas
- *Dataset Size*
- Davies-Boudin
- Tempo
- *Silhouette*
- Métricas quantitativas
- Calinski-Harabaz

Fonte: Elaborada pelo autor.

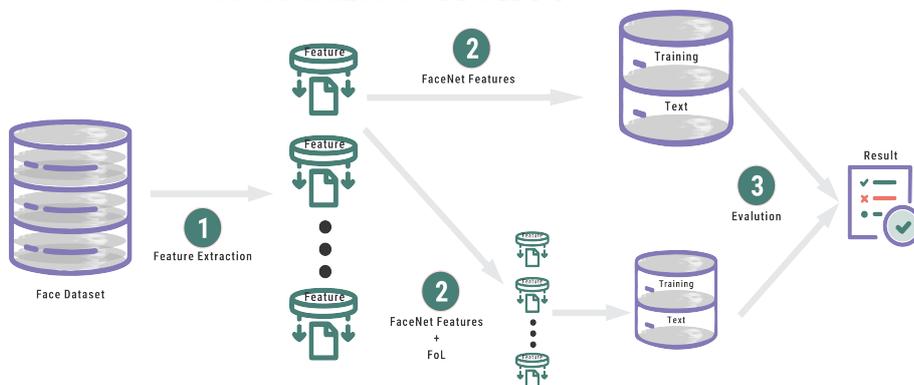
4.1 Avaliação do FoL em *array*

Para avaliar a eficácia e robustez do método proposto, foram utilizados 4 conjuntos de dados amplamente usados em artigos de reconhecimento facial: AR (MARTINEZ; BENAVENTE, 1998), Extended Yale B (GEORGHIADES *et al.*, 2001), LFW (HUANG *et al.*, 2007) e CelebA (LIU *et al.*, 2015). Para cada *dataset*, a extração de características foi realizada com o extrator FaceNet, e então o algoritmo FoL foi aplicado para compressão de características usando 1.0004907000000118 para base logarítmica e 50 para Bias.

Esse valor de base foi obtido após 20 minutos de treinamento com o *dataset* AR usando a parte de treinamento. Os experimentos mostram que mesmo mudando o *dataset* de treinamento e o algoritmo extrator, esse valor não muda muito. Após a compressão FoL, cada *dataset* foi serializado e salvo no armazenamento do computador para obter o tamanho total dessas características comprimidas.

Os classificadores SVM e k-NN foram aplicados usando a divisão completa do *dataset*, conforme descrito nas seções seguintes, para obter a precisão usando características descompactadas (FaceNet) e compactadas (FaceNet+FoL). Para cada experimento a seguir, o tempo médio gasto para classificar um rosto não difere entre características comprimidas e não comprimidas, cerca de 0,0006 ms. A Figura 17 apresenta o processo utilizado para gerar os resultados.

Figura 17 – Avaliação da abordagem proposta. Para cada *dataset* de face, o FaceNet foi aplicado para extrair *features* na etapa 1. Na etapa 2, o FoL foi aplicado no *dataset* para cada *array* de *feature*, gerando *features* compactadas. Esses *datasets* foram salvos após esta etapa para avaliar o tamanho final do conjunto de dados. Finalmente, a etapa 3 avalia cada *datasetss* usando SVM e K-NN.



Fonte: Elaborada pelo autor.

4.1.1 Resultados experimentais no dataset AR

Para avaliar o método, foi utilizada apenas a primeira sessão de cada pessoa. Apenas as quatro primeiras imagens de cada indivíduo foram usadas para treinamento, que representam as imagens com diferentes expressões faciais: expressão neutra, sorriso, raiva e grito. Todas as seis imagens com oclusão, três imagens com óculos e três imagens com lenço foram usadas para teste. As outras imagens não foram consideradas.

Os testes foram divididos entre as imagens de teste com lenço(*Scarf*) e as imagens de teste com óculos(*Glasses*), como mostra a Tabela ???. Analisando a Tabela ???, observamos que o método proposto reduz o tamanho do arquivo, que vai de 280,7 kB sem compressão para 32,5 kB após a compressão das características do FaceNet.

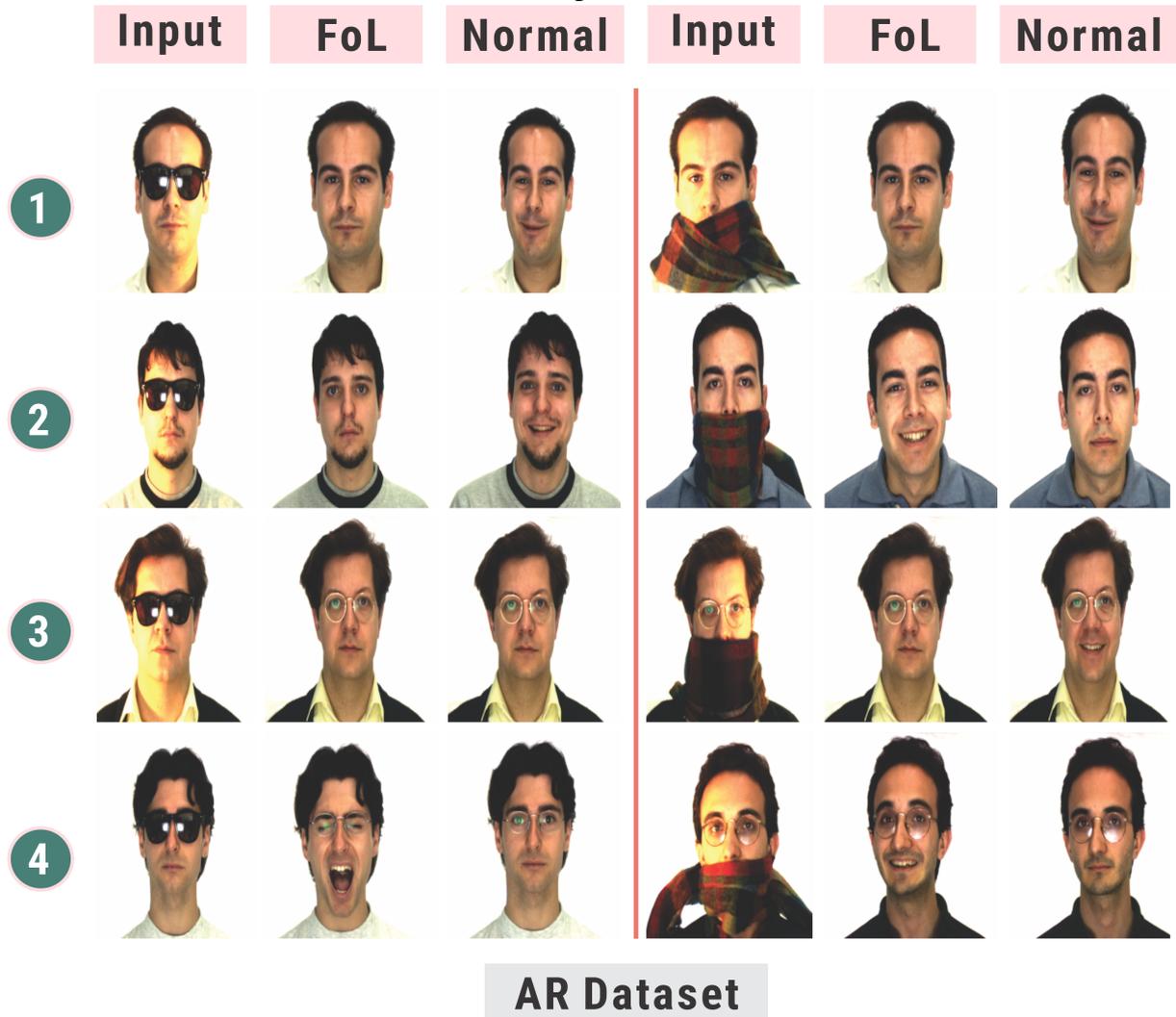
Além disso, podemos ver que a compressão não apenas diminui o tamanho do arquivo, mas também aumenta a precisão obtida ao usar SVM como classificador para testes de lenço. A Figura 18 apresenta alguns exemplos de resultados de correspondência de rosto para o *dataset* AR.

Tabela 1 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no *dataset* AR com lenço e óculos como objetos de oclusão. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.

AR - Lenço e Óculos						
Dataset	Classificador	FoL	Acc (%)	Tempo (s)	Tamanho	Redução (%)
Lenço	SVM	Não	93.5	0.16	280.7 kB	-
	k-NN	Não	94.75	0.046	280.7 kB	-
	SVM	Sim	96.75	0.15	32.5 kB	88.42
	k-NN	Sim	95.00	0.0467	32.5 kB	88.42
Óculos	SVM	Não	77.28	0.13	280.7 kB	-
	k-NN	Não	81.97	0.045	280.7 kB	-
	SVM	Sim	81.23	0.15	32.5 kB	88.42
	k-NN	Sim	81.48	0.043	32.5 kB	88.42

Fonte: Elaborada pelo autor.

Figura 18 – Figura com resultados de correspondência de rostos para o *dataset* AR. Mesmo com a presença de diferentes tipos de oclusão, lenço e óculos, o FoL obtém resultados diferentes, mas mantém a mesma pessoa relacionada ao rosto.



Fonte: Elaborada pelo autor.

4.1.2 Resultados experimentais no *dataset* LFW

Uma vez que as imagens no *dataset* LFW foram tiradas em um ambiente não controlado, as imagens não possuem um padrão consistente e a maioria das imagens não possui rostos alinhados, o que impacta nos valores de precisão. No entanto, o objetivo é evidenciar a capacidade de compressão do método proposto sem perder a precisão que seria obtida pelo método de extração não comprimido. A Tabela 2 apresenta os resultados obtidos para o *dataset* LFW.

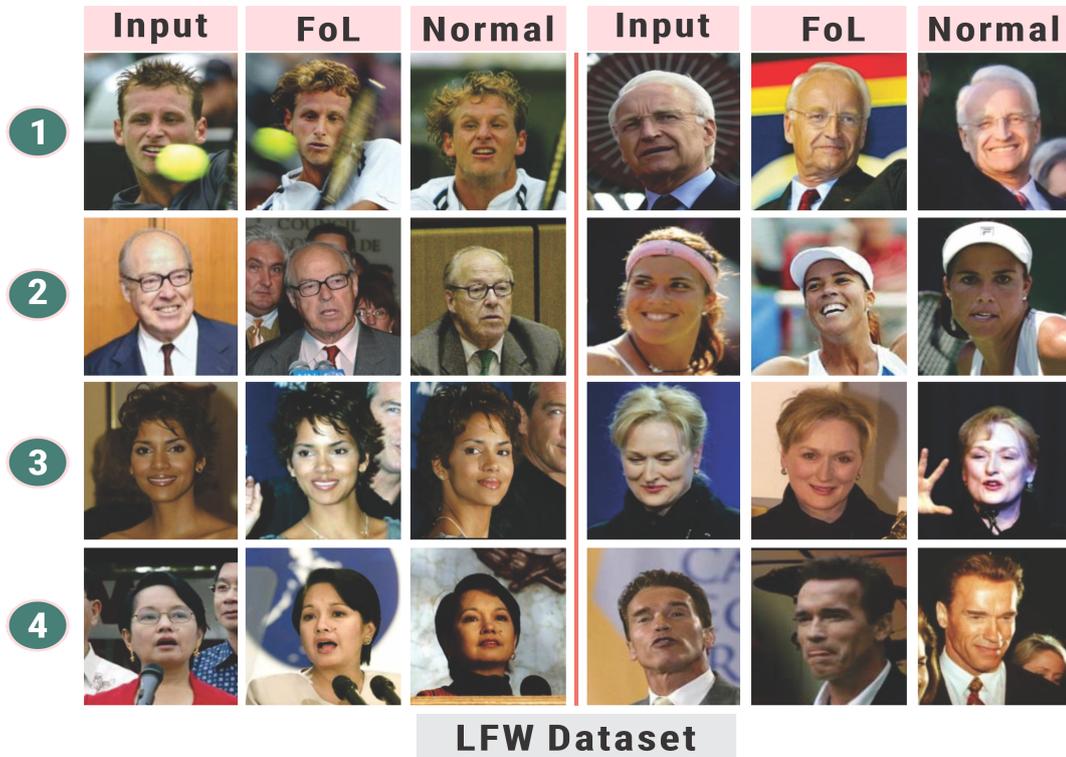
Tabela 2 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no *dataset* LFW. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.

Dataset	Classificador	FoL	Acc (%)	Tempo (s)	Tamanho	Redução (%)
LFW	SVM	Não	95.71	1.213	1.6 MB	-
	k-NN	Não	94.73	0.71	1.6 MB	-
	SVM	Sim	94.59	1.016	186.2 kB	88.63
	k-NN	Sim	95.92	0.59	186.2 kB	88.63

Fonte: Elaborada pelo autor.

Mais uma vez, a precisão obtida aumentou quando o método proposto foi aplicado às características extraídas usando o classificador k-NN, alcançando um aumento de 1,19% com a CNN como detector de rostos. Além disso, a redução do tamanho do arquivo das características extraídas pelo FaceNet após a compressão foi de 88,63% para rostos detectados, respectivamente. A Figura 19 apresenta alguns exemplos de resultados de correspondência de rostos para o *dataset* LFW.

Figura 19 – Comparação dos resultados das *features* compactados e descompactados considerando a precisão de correspondência facial obtida por cada classificador como métrica de avaliação no conjunto de dados LFW. Também é apresentado o tamanho de cada arquivo e a redução percentual após a compactação.



Fonte: Elaborada pelo autor.

4.1.3 Resultados experimentais no dataset EYale B

A Tabela 3 apresenta os resultados obtidos a partir do banco de dados Extended Yale B. A redução mais significativa no tamanho do arquivo antes e depois da compressão foi alcançada no conjunto de dados Extended Yale B. O arquivo comprimido mostrou uma redução de 91,3% no conjunto de características original. A compressão também melhorou todos os resultados de precisão neste banco de dados, confirmando não apenas a capacidade do método de comprimir, mas também afirma a capacidade do método de reforçar a descrição dos rostos dada pelas características extraídas pelo FaceNet. A Figura 20 apresenta alguns exemplos de resultados de correspondência de rostos para o conjunto de dados Extended Yale B.

Tabela 3 – Comparação dos resultados das características comprimidas e não comprimidas, considerando a precisão de correspondência de rostos obtida por cada classificador como métrica de avaliação no conjunto de dados Extended Yale B. O tamanho de cada arquivo e a porcentagem de redução após a compressão também são apresentados.

Dataset	Classificador	FoL	Acc (%)	Tempo (s)	Tamanho	Redução (%)
Extended Yale B	SVM	Não	95.92	2.6	13.8 MB	-
	k-NN	Não	96.94	3.89	13.8 MB	-
	SVM	Sim	96.89	1.52	1.2 MB	91.30
	k-NN	Sim	97.06	4.02	1.2 MB	91.30

Fonte: Elaborada pelo autor.

Figura 20 – Resultados da Face Matching para o conjunto de dados Extended Yale B. Mesmo com diferentes iluminações, uma questão particular deste conjunto de dados, o FoL alcança a mesma pessoa relacionada a Face.



Fonte: Elaborada pelo autor.

4.1.4 Resultados experimentais no dataset CelebA

O CelebA não apresenta um rótulo para cada rosto em termos de reconhecimento facial. Os rótulos deste conjunto de dados estão relacionados às informações de emoções e

gênero. Assim, apenas a extração de características, usando o FaceNet, e a compressão, usando o FoL, foram realizadas para verificar o tamanho que uma grande base de imagens pode ter e a redução que o método proposto pode executar.

Este foi o maior banco de dados ao qual tivemos acesso. No entanto, é importante notar que muitas redes de reconhecimento facial usam muito mais do que 200.000 imagens. DeepFace e FaceNet, por exemplo, usaram 8M e 200M de imagens na etapa de treinamento, respectivamente.

A Tabela 4 apresenta os resultados obtidos a partir do banco de dados CelebA. O tamanho do arquivo de características diminuiu de 226,1 MB para 30,7 MB com a CNN como detector de rostos. Essa redução representa 86,28% dos arquivos originais usando CNN como detector facial. Isso representa uma redução significativa ao considerar que esses arquivos serão compartilhados entre sistemas e dispositivos IoT. A Figura 21 apresenta alguns exemplos de resultados de correspondência de rostos para o conjunto de dados CelebA.

Tabela 4 – Comparison of the file size of compressed and uncompressed features in CelebA dataset.

CelebA		
Detection algorithm	Tamanho	Redução (%)
Normal		
CNN	226.1 MB	-
FoL		
CNN	30.7 MB	86.42

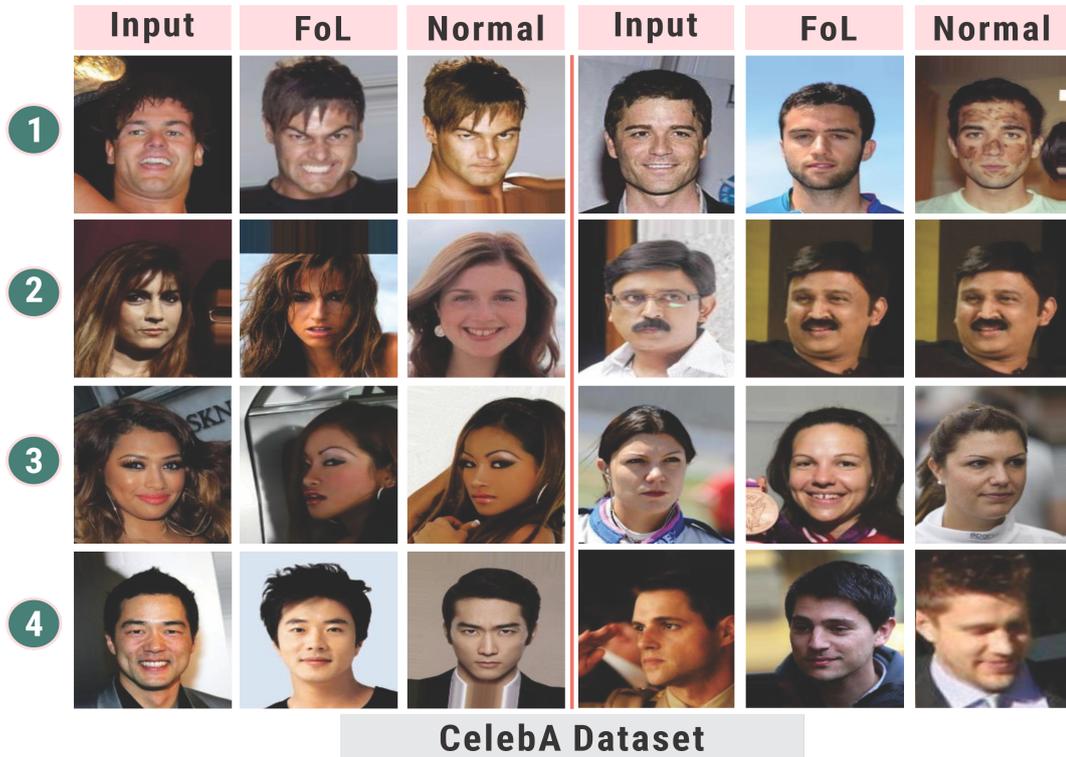
Fonte: Elaborada pelo autor.

4.2 Avaliação do FoL em Imagens

Considerando as etapas apresentadas na Figura 22 e as imagens do *dataset* de tomografia computadorizada, esta seção é dividida em 2 blocos principais: primeiro, uma apresentação da avaliação da nova abordagem proposta para a segmentação de pulmão 3D, seguida por uma segunda etapa mostrando uma comparação com métodos de relacionados à segmentação de pulmão 3D.

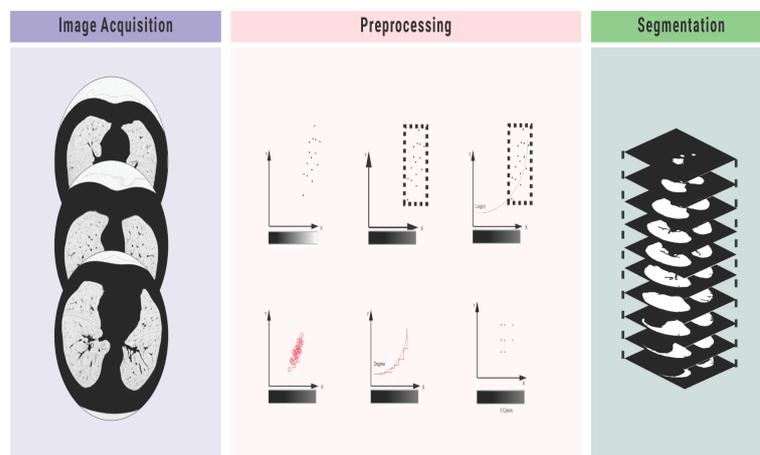
Mostrando cada iteração do processo de aprendizado, a Figura 23 apresenta a precisão média após 10 validações cruzadas no processo de aprendizado do algoritmo FoL, que precisa

Figura 21 – Comparação do tamanho do arquivo das *features* compactados e não compactados no CelebA conjunto de dados



Fonte: Elaborada pelo autor.

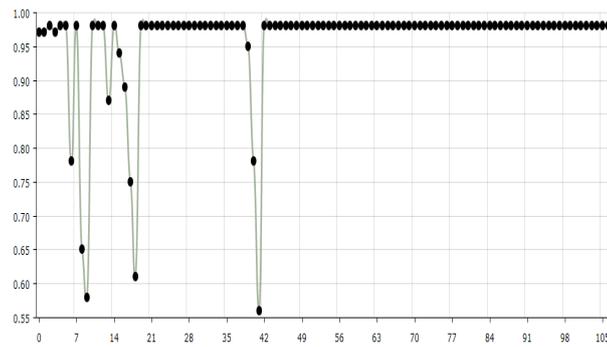
Figura 22 – Metodologia proposta. Nas etapas de aquisição de imagens, as imagens pulmonares do *data-set* de tomografia computadorizada são inicialmente armazenadas e separadas por suas classes. A coluna Pré-processamento apresenta o novo método de agrupamento proposto e a etapa de Segmentação na terceira coluna apresenta a segmentação da imagem e a construção da visualização 3D.



Fonte: Elaborada pelo autor.

treinar sua base logarítmica para se ajustar melhor às regiões de agrupamento.

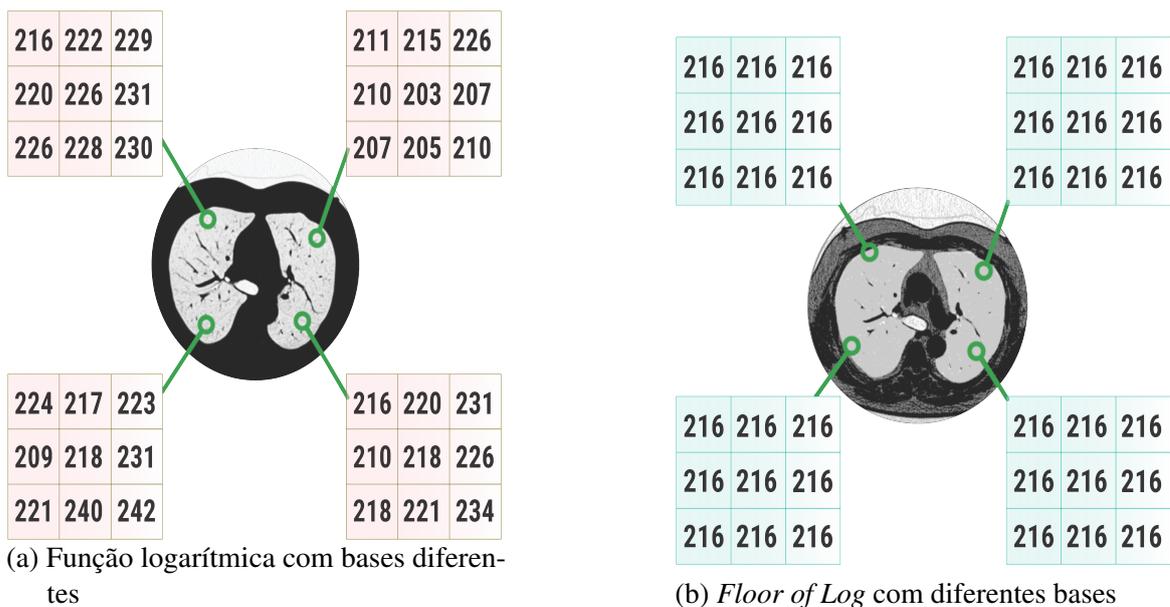
Figura 23 – Avaliação do processo de aprendizado pelo algoritmo de treinamento proposto com 10 cross-validation usando 105 iterações. A cada iteração, o algoritmo muda a base 'b' usada em FoL.



Fonte: Elaborada pelo autor.

Para abordagem experimental, a Figura 10(b) indica um valor único de nível de cinza e alguma presença de buracos que constituem os vasos no *dataset* LUNA16. Esses vasos são facilmente e rapidamente removidos usando o operador morfológico flood-fill.

Figura 24 – Efeitos do FoL no *dataset* LUNA16. Os valores originais apresentados na imagem (a) foram convertidos no valor único da região aerada (b) após o método FoL.

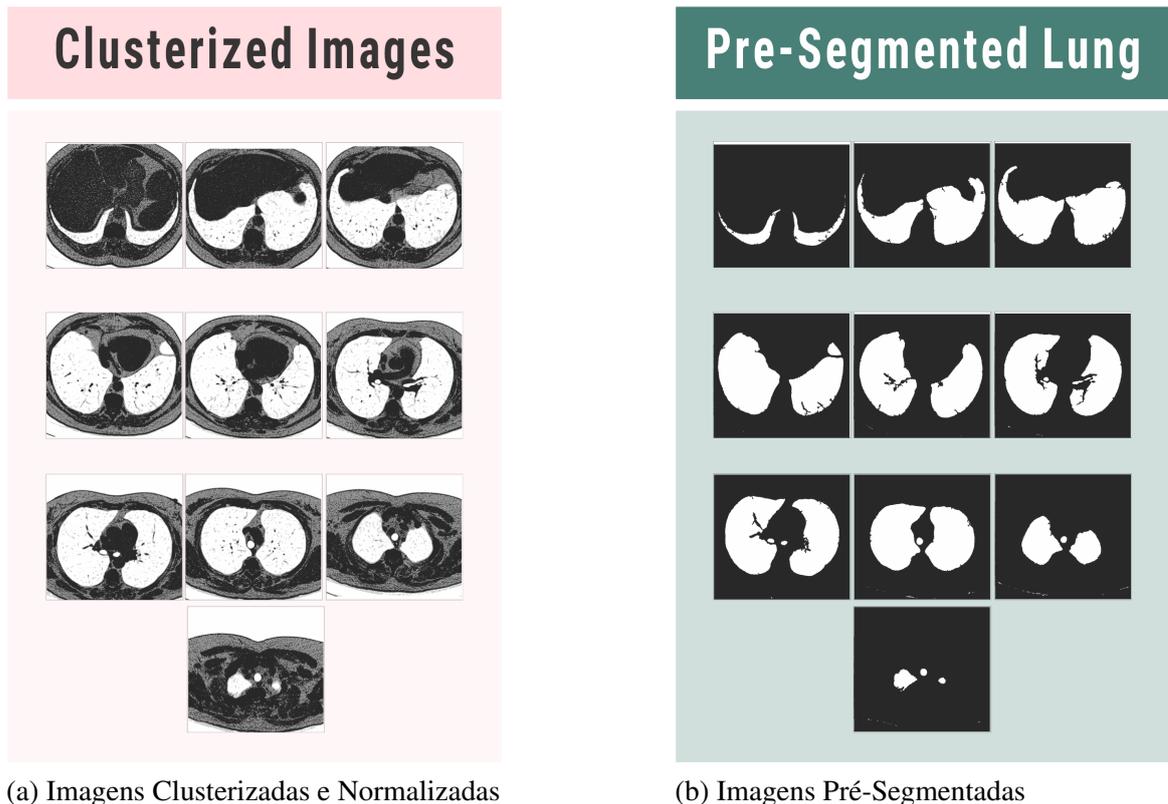


Fonte: Elaborada pelo autor.

O processo pré-segmentado é apresentado na Figura 25, onde o valor do pixel das

regiões agrupadas é usado para extrair os componentes aerados. A Figura 25a apresenta os resultados da região de agrupamento usando o algoritmo FoL. Usando essa abordagem, a região aerada foi extraída selecionando o valor do pixel correspondente a essa área, conforme mostrado na Figura 25b.

Figura 25 – Processo de agrupamento no dataset LAPISCO. Após o algoritmo FoL, a região pulmonar foi extraída por limiarização a partir de um único valor de pixel.



(a) Imagens Clusterizadas e Normalizadas

(b) Imagens Pré-Segmentadas

Fonte: Elaborada pelo autor.

Finalmente, usando todos os exames da fatia, o pulmão foi extraído em forma tridimensional, selecionando os 2 maiores volumes, essa abordagem é necessária para remover artefatos eventuais, como alguns gases indesejados e a traqueia. O resultado desta operação é apresentado na Figura 26.

As Tabelas 5, 6, 7 apresentam o Tempo Médio da tarefa de segmentação, a análise de similaridade e os resultados qualitativos, respectivamente. Em todos os resultados, o método FoL foi comparado com métodos normalmente utilizados nesta tarefa de segmentação: 3D Adaptive Crisp ACM (REBOUÇAS *et al.*, 2017b), o algoritmo de crescimento de região 3D automático (3D RG), o algoritmo de conjunto de nível baseado no método de propagação coerente (LSCPM) (WANG *et al.*, 2011), (WANG *et al.*, 2014) e a segmentação semiautomática realizada por um

Figura 26 – Exemplos de segmentação pulmonar. Após a extração dos 2 maiores volumes na combinação de todo o exame, o processo de segmentação é finalizado.



Fonte: Elaborada pelo autor.

especialista usando a caixa de ferramentas 3D OsiriX (EUOT) (HUNTER-SMITH *et al.*, 2015).

A performance foi apresentada na Tabela 5. O método LSCPM alcança 235 segundos em média no tempo da tarefa de segmentação, seguido por 334 segundos com o RG 3D. A abordagem do modelo ACM 3D gastou 441 segundos, um pouco mais, já que usa uma abordagem de aprendizado de máquina interna para analisar a vizinhança. O EUOT gastou o maior tempo, o que era esperado, já que é um modelo dependente do usuário. A abordagem FoL reduz seu valor com um enorme ganho de velocidade, alcançando apenas 19 segundos em média no tempo.

Tabela 5 – Tempo médio de segmentação do FoL em relação a outros métodos.

Method	Time (s)
FoL	19 ±1.12
ACM 3D	441 ±0,11
3D RG	334 ±0,16
EUOT	643 ±0,11
LSCPM	235 ±0,11

Fonte: Elaborada pelo autor.

A Tabela 6 apresenta os resultados relativos à similaridade. Note que houve um crescimento substancial do FoL em comparação com as outras abordagens nas métricas, indicando bons valores de segmentação quando comparados ao resultado de um especialista.

Na Tabela 7, os valores de Sensibilidade, Precisão e MCC são apresentados como

Tabela 6 – Índices de similaridade com FoL em relação a outros métodos de segmentação

Method	Similarity Indexes		
	HD	DICE	Jaccard
FoL	3.51±0.31	83.63 ±0,193	99.73 ±0,133
ACM 3D	3.31±0.23	83.22 ±0,282	99.78 ±0,152
3D RG	3.71 ±0.63	82.33 ±0,184	99.66 ±0,141
EUOT	3.74±0.74	82.95 ±0,231	99.61 ±0,124
LSCPM	3.62±0.25	80.91 ±0,172	99.43 ±0,212

Fonte: Elaborada pelo autor.

medidas qualitativas. Nesse sentido, é avaliada a capacidade do método de corrigir ou não as regiões relacionadas ao pulmão e não a similaridade do método em relação ao de um especialista. Pode-se ver que o FoL alcança valores quase iguais aos dos outros métodos, com resultados ligeiramente melhores.

Tabela 7 – Índices quantitativos para FoL em relação a outros métodos de segmentação

Method	Quantitative indexes		
	Se	Acc	MCC
FoL	83.87±0,066	99.62±0,046	83.08±0,192
ACM 3D	82.71±0,131	99.78±0,129	83.40±0,320
3D RG	83.01±0,510	99.66±0,361	82.79±0,645
EUOT	83.62±0,513	99.73±0,342	82.37±0,514
LSCPM	84.81±0,447	99.43±0,281	80.34±0,423

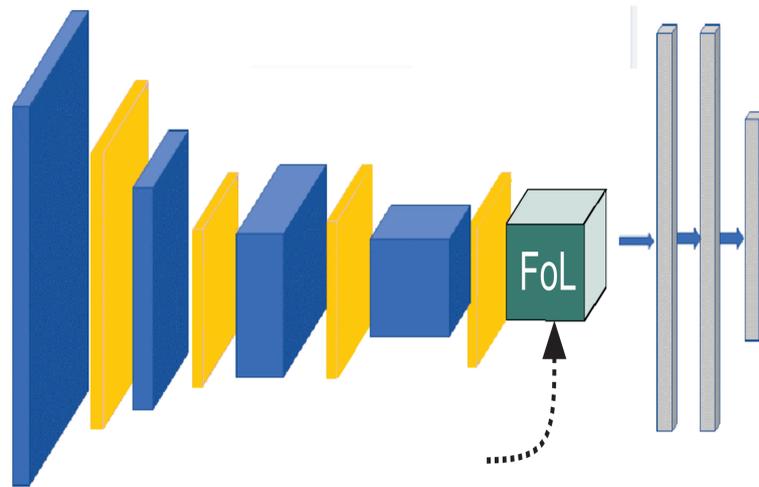
Fonte: Elaborada pelo autor.

4.3 Avaliação do FoL dentro de modelos de Deep Learning

Nesta seção, apresentamos os resultados do uso do FoL aplicado aos modelos VGG16, GoogLeNet e Attention nos conjuntos de dados CIFAR-10 e CIFAR-100. Para facilitar a comparação, as Tabelas 8,9 mostram a acurácia de cada modelo e uma análise de agrupamento é feita nas Tabelas 10, 11, e 12, considerando a presença e a ausência do FoL, como também a comparação com um método linear. A transformada FoL foi adicionada nas redes após o processo da extração de características de cada uma, antes de serem enviados para as camadas de classificação, anterior ao achatamento das features, conforme ilustrado na Figura 27.

As Figuras 28, 29 e 30 correspondem ao processo de aprendizado do FoL em comparação com os métodos tradicionais para os modelos VGG16, GoogLeNet e Attention, respectivamente. Essas figuras mostram as curvas de treinamento dos modelos ao longo do tempo. Podemos observar que, embora haja algumas flutuações, o desempenho do FoL é comparável aos

Figura 27 – Modelo geral de adição do FoL como camada dentro dos modelos.



Fonte: Elaborada pelo autor.

métodos tradicionais, o que indica que o FoL é eficaz no processo de aprendizado dos modelos.

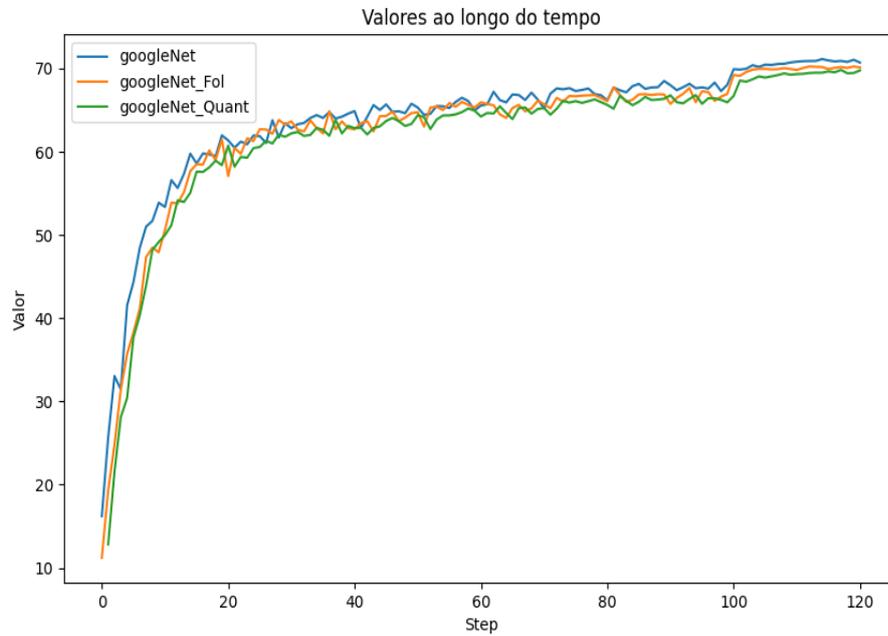
Figura 28 – Processo de aprendizado do FoL em comparação com os demais métodos na rede VGG



Fonte: Elaborada pelo autor.

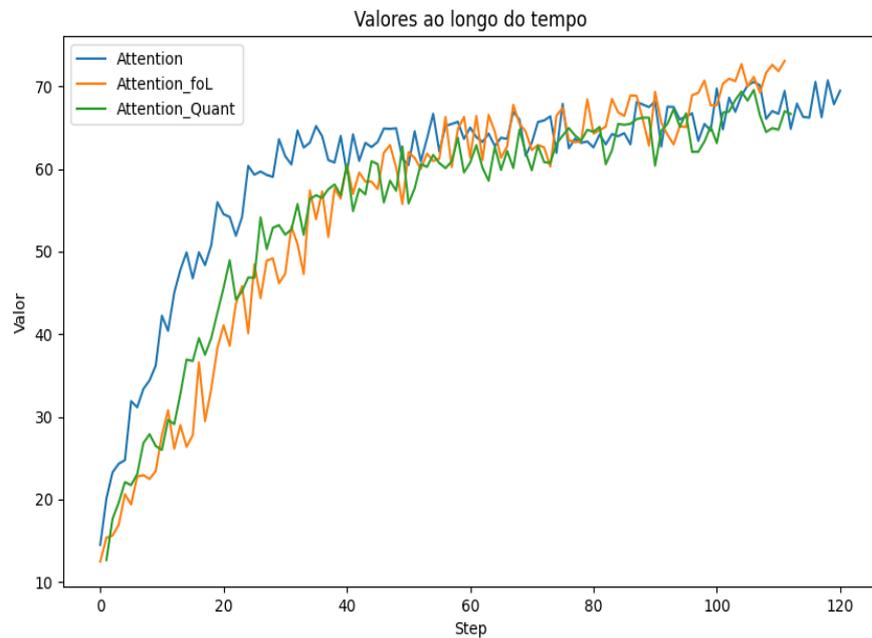
Na Tabela 8, são apresentadas as acurácias dos modelos VGG16, GoogLeNet e Attention no conjunto de dados CIFAR-10. Os resultados indicam que, com o uso do FoL, houve uma ligeira queda na acurácia para todos os modelos em comparação com o treinamento sem FoL. No entanto, essa diferença é pequena e dentro de uma faixa relativamente, demonstrando que o FoL não compromete significativamente o desempenho dos modelos.

Figura 29 – Processo de aprendizado do FoL em comparação com os demais métodos na rede Googlenet



Fonte: Elaborada pelo autor.

Figura 30 – Processo de aprendizado do FoL em comparação com os demais métodos na rede Attention



Fonte: Elaborada pelo autor.

Tabela 8 – Acurácia dos modelos no conjunto de dados CIFAR-10.

Modelo	VGG16	GoogLeNet	Attention
Com FoL	91.4%	90.2%	90.3%
Quant	91.4%	90.1%	90.4%
Sem FoL	91.8%	92.1%	91.4%

Fonte: Elaborada pelo autor.

A Tabela 9 mostra as acurácias dos modelos VGG16, GoogLeNet e Attention no conjunto de dados CIFAR-100.

Tabela 9 – Acurácia dos modelos no conjunto de dados CIFAR-100.

Modelo	VGG16	GoogLeNet	Attention
Com FoL	70.1%	71.1%	72.4%
Quant	68.5%	70.4%	69.8%
Sem FoL	66.4%	72.1%	70.2%

Fonte: Elaborada pelo autor.

Nas Tabelas 10, 11, e 12,, apresentamos os resultados da análise de agrupamento utilizando as métricas de Davies-Bouldin (DB), Calinski-Harabasz (CH) e Silhouette (S) para os conjuntos de dados CIFAR-100.

Tabela 10 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando VGG16.

Métrica	Com FoL	Sem FoL	Quant
DB (CIFAR-100)	1.74	1.59	1.58
CH (CIFAR-100)	137.01	37.65	40.08
S (CIFAR-100)	0.17	0.20	0.17

Fonte: Elaborada pelo autor.

Tabela 11 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando GoogleNet.

Métrica	Com FoL	Sem FoL	Quant
DB (CIFAR-100)	1.85	1.95	1.97
CH (CIFAR-100)	73.67	55.61	41.14
S (CIFAR-100)	0.07	0.10	0.10

Fonte: Elaborada pelo autor.

Tabela 12 – Análise de agrupamento nos conjuntos de dados CIFAR-100 utilizando Attention.

Métrica	Com FoL	Sem FoL	Quant
DB (CIFAR-100)	1.31	1.73	1.41
CH (CIFAR-100)	124	32.65	45.08
S (CIFAR-100)	0.12	0.15	0.13

Fonte: Elaborada pelo autor.

Ao analisar os resultados, é possível observar que o uso do FoL apresentou um aumento significativo no índice de Calinski-Harabasz, cerca de 3 vezes maior quando comparado aos outros métodos, tanto para o conjunto de dados CIFAR-100. Este aumento expressivo no

índice de Calinski-Harabasz sugere que o FoL pode contribuir para uma melhor separação e coesão dos agrupamentos nos dados.

Como o índice Davies-Bouldin e a medida de Silhueta apresentaram valores similares para os experimentos com e sem FoL, isso sugere que a adição do FoL não afetou negativamente a qualidade dos agrupamentos em termos de dispersão dentro do agrupamento e separação entre os agrupamentos e a consistência dos agrupamentos.

No entanto, como mencionado anteriormente, o índice CH mostrou uma melhora significativa com a adição do FoL. Isso indica que, embora as métricas DB e S não apresentem diferenças significativas, a aplicação do FoL ainda é capaz de proporcionar agrupamentos mais densos e melhor separados, como evidenciado pelo aumento no índice CH.

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o algoritmo FoL, um método de transformação de dados com capacidade de efetuar operações que resultam em efeitos de agrupamento e compressão de dados. A aplicação do FoL em diversos modelos de aprendizado de máquina, incluindo VGG16, GoogLeNet e Attention, mostrou resultados promissores tanto em termos de acurácia quanto de eficiência computacional.

Nos experimentos realizados, o FoL foi aplicado em diferentes conjuntos de dados, incluindo LUNA16, LAPISCO, AR, LFW, EYaleB, CelebA, Cifar10 e Cifar100. Em todos os casos, o FoL demonstrou ser eficaz na compressão de dados e na melhoria da precisão dos modelos.

Como destaque para compressão, no conjunto de dados Extended Yale B, a aplicação do FoL resultou em uma redução significativa de 91,3% no tamanho do conjunto de características original. Além disso, a compressão também melhorou todos os resultados de precisão neste banco de dados, confirmando não apenas a capacidade do método de comprimir, mas também afirma a capacidade do método de reforçar a descrição dos rostos dada pelas características extraídas pelo FaceNet.

Um ponto bastante interessante do ponto de vista de clusterização em imagens, o método conseguiu atingir resultados similares aos das aplicações vigentes. Entretanto, um ponto significativo do FoL está na velocidade com que esses resultados foram atingidos: em média 19 segundos ao contrário dos demais que superam os 3 minutos.

No entanto, é importante notar que os parâmetros específicos podem variar dependendo do experimento específico ou do modelo utilizado. Além disso, embora os resultados sejam promissores, são necessárias mais pesquisas para explorar plenamente o potencial do algoritmo FoL em diferentes aplicações e contextos.

5.0.1 *Trabalhos Futuros*

Do ponto de vista de imagens e, uma vez que o método FoL possui características de clusterização, seria interessante a avaliação do mesmo como um descritor de borda e, portanto, associar essa característica com a CNN para entender seus impactos.

Dessa forma, o FoL poderá ser avaliado como um otimizador no processo de interpretação inicial do modelo, ou mesmo auxiliar como um down-sample de toda a camada

de extração. Espera-se que essa atuação implique na redução de parâmetros necessários dos modelos presentes na literatura, tornando os modelos conseqüentemente mais rápidos.

REFERÊNCIAS

- AL-MAITAH, M.; ALZUBI, A. A.; ALARIFI, A. An optimal storage utilization technique for iot devices using sequential machine learning. **Computer Networks**, London, v. 152, p. 98–105, 2019.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. *In*: IEEE. **2017 International Conference on Engineering and Technology (ICET)**. Antalya, Turkey: IEEE, 2017. p. 1–6.
- ALLMUTTAR, A. Y.; ALKHAFI, S. K. Using data mining techniques deep analysis and theoretical investigation of covid-19 pandemic. **Measurement: Sensors**, London, v. 27, n. 3, p. 1–6, 2023.
- ALVES, S. A.; REBOUÇAS, E. S.; OLIVEIRA, S. F.; BRAGA, A. M.; FILHO, P. P. R. Lung diseases classification by analysis of lung tissue densities. **IEEE Latin America Transactions**, IEEE, v. 18, n. 9, p. 1329–1336, 2018.
- AULAKH, K.; ROUL, R. K.; KAUSHAL, M. E-learning enhancement through educational data mining with covid-19 outbreak period in backdrop: A review. **International Journal of Educational Development**, London, v. 101, n. 4, p. 1–17, 2023.
- AZAR, J.; MAKHOUL, A.; BARHAMGI, M.; COUTURIER, R. An energy efficient iot data compression approach for edge machine learning. **Future Generation Computer Systems**, London, v. 96, p. 168–175, 2019.
- BALIARSINGH, S.; PANDA, P. K.; MOHANTY, M. N. Unsupervised technique for cardiac data compression with β wavelet transform for telemedicine. **Measurement: Sensors**, London, v. 26, n. 7, p. 1–7, 2023.
- CHOUHAN, V.; SINGH, S. K.; KHAMPARIA, A.; GUPTA, D.; TIWARI, P.; MOREIRA, C.; DAMAŠEVIČIUS, R.; ALBUQUERQUE, V. H. C. D. A novel transfer learning based approach for pneumonia detection in chest x-ray images. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 2, p. 559, 2020.
- CRUZ, M. A. A. D.; RODRIGUES, J. J. P. C.; AL-MUHTADI, J.; KOROTAIEV, V. V.; ALBUQUERQUE, V. H. C. D. A reference model for internet of things middleware. **IEEE Internet of Things Journal**, v. 5, n. 2, p. 871–883, 2018.
- DICE, L. R. Measures of the amount of ecologic association between species. **Ecology**, Wiley Online Library, v. 26, n. 3, p. 297–302, 1945.
- DOL, S. M.; JAWANDHIYA, P. M. Classification technique and its combination with clustering and association rule mining in educational data mining—a survey. **Engineering Applications of Artificial Intelligence**, London, v. 122, n. 3, p. 10–15, 2023.
- DOMINIC, N.; ELWIREHARDJA, G. N.; PARDAMEAN, B. Data mining for the global multiplex weekly average income analysis. **Procedia Computer Science**, London, v. 219, p. 52–59, 2023.
- FANG, J.; YUAN, Y.; LU, X.; FENG, Y. Muti-stage learning for gender and age prediction. **Neurocomputing**, London, v. 334, n. 2, p. 114–124, 2019.

FANG, R.; GUPTA, A.; HUANG, J.; SANELLI, P. Tender: Tensor non-local deconvolution enabled radiation reduction in ct perfusion. **Neurocomputing**, London, v. 229, p. 13–22, 2017.

FARAHZADI, A.; SHAMS, P.; REZAZADEH, J.; FARAHBAKHSR, R. Middleware technologies for cloud of things: a survey. **Digital Communications and Networks**, London, v. 4, n. 3, p. 176–188, 2018.

FÉLIX, J. H. da S. **Métodos de Contornos Ativos Hilbert 2D na segmentação de imagens dos pulmões em Tomografia Computadorizada do Tórax. 2011.** Tese (Doutorado) – Universidade Federal do Ceará, 2011. Programa de pós-graduação em engenharia de teleinformática.

GEORGHIADES, A. S.; BELHUMEUR, P. N.; KRIEGMAN, D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 23, n. 6, p. 643–660, 2001.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. New Jersey: pearson prentice hall, 2017. v. 4. 1-1192 p. ISBN 9780133356724.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Massachusetts: MIT Press, 2016. v. 1. 1-775 p.

GUIMARAES, R. R.; PASSOS, L. A.; FILHO, R. H.; ALBUQUERQUE, V. H. C. d.; RODRIGUES, J. J. P. C.; KOMAROV, M. M.; PAPA, J. P. Intelligent network security monitoring based on optimum-path forest clustering. **IEEE Network**, v. 33, n. 2, p. 126–131, 2019.

GUMAEI, A.; HASSAN, M. M.; HASSAN, M. R.; ALELAIWI, A.; FORTINO, G. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. **IEEE Access**, IEEE, v. 7, p. 36266–36273, 2019.

GUPTA, N.; GUPTA, D.; KHANNA, A.; FILHO, P. P. R.; ALBUQUERQUE, V. H. C. de. Evolutionary algorithms for automatic lung disease detection. **Measurement**, London, v. 140, p. 590–608, 2019.

HAJIMANI, E.; RUANO, M.; RUANO, A. An intelligent support system for automatic detection of cerebral vascular accidents from brain ct images. **Computer methods and programs in biomedicine**, London, v. 146, p. 109–123, 2017.

HAN, G.; LIU, X.; ZHANG, H.; ZHENG, G. W.; SOOMRO, N. Q.; WANG, M.; LIU, W. Hybrid resampling and multi-feature fusion for automatic recognition of cavity imaging sign in lung ct. **Future Gener. Comput. Syst.**, v. 99, p. 558–570, 2019.

HASSAN, M. M.; ALAM, M. G. R.; UDDIN, M. Z.; HUDA, S.; ALMOGREN, A.; FORTINO, G. Human emotion recognition using deep belief network architecture. **Information Fusion**, London, v. 51, p. 10–18, 2019.

HAUSDORFF, F. Bemerkung über den inhalt von punktmengen. **Mathematische Annalen**, Springer, v. 75, n. 3, p. 428–433, 1914.

HAYKIN, S. **Neural Networks and Learning Machines**. London: Pearson, 2009. 1-936 p.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *In: IEEE. Proceedings of the IEEE international conference on computer vision*. Santiago, Chile: IEEE, 2015. p. 1026–1034.

HE, M.; ZHAO, Q.; ZHANG, H.; KANG, C.; LI, W.; HAN, M. Graph representation learning via redundancy reduction. *Neurocomputing*, London, v. 533, p. 161–177, 2023. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231223002102>. Acesso em: 15 jun. 2023.

HUANG, G. B.; RAMESH, M.; BERG, T.; LEARNED-MILLER, E. **Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments**. Massachusetts, 2007. Disponível em: <https://vis-www.cs.umass.edu/lfw/>. Acesso em: 9 dec. 2022.

HUNTER-SMITH, D.; ALEXANDRA, R.; SPYCHAL, R.; MICHAEL, P. 3d volumetric analysis and haptic modeling for preoperative planning in breast reconstruction. *Anaplastology*, v. 4, n. 138, p. 2161–1173, 2015.

HUSSAIN, T.; MUHAMMAD, K.; SER, J. D.; BAIK, S. W.; ALBUQUERQUE, V. H. C. d. Intelligent embedded vision for summarization of multi-view videos in iiot. *IEEE Transactions on Industrial Informatics*, v. 16, n. 4, p. 1–1, 2019.

HUSSAIN, T.; MUHAMMAD, K.; ULLAH, A.; CAO, Z.; BAIK, S. W.; ALBUQUERQUE, V. H. C. de. Cloud-assisted multi-view video summarization using cnn and bi-directional lstm. *IEEE Transactions on Industrial Informatics*, v. 16, n. 1, p. 1–1, 2019.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In: Proceedings of the 32nd International Conference on International Conference on Machine Learning*. Lille, France: ACM, 2015. p. 448–456.

ISLAM, M. T.; BASAK, P. K.; BHOWMIK, P.; KHAN, M. Data clustering using hybrid genetic algorithm with k-means and k-medoids algorithms. *In: IEEE. 2019 23rd International Computer Science and Engineering Conference (ICSEC)*. Phuket, Thailand: IEEE, 2019. p. 123–128.

JACKSON, M. **Software Requirements & Specifications: a lexicon of practice, principles and prejudices**. Boston: ACM Press/Addison-Wesley Publishing Co., 1995. ISBN 978-0201877120.

JHA, C. K.; KOLEKAR, M. Electrocardiogram data compression techniques for cardiac healthcare systems: A methodological review. *IRBM*, London, London, v. 43, n. 3, p. 217–228, 2022.

KAYA, H.; GÜRPINAR, F.; SALAH, A. A. Video-based emotion recognition in the wild using deep transfer learning and score fusion. *Image and Vision Computing*, London, v. 65, p. 66–75, 2017.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015.

LI, Q.; XIONG, Q.; JI, S.; WEN, J.; GAO, M.; YU, Y.; XU, R. Using fine-tuned conditional probabilities for data transformation of nominal attributes. *Pattern Recognition Letters*, London, v. 128, p. 107–114, 2019.

- LIU, Z.; BARAHONA, M. Graph-based data clustering via multiscale community detection. **Applied Network Science**, Berlin, v. 5, n. 3, p. 1–20, 2020.
- LIU, Z.; LUO, P.; WANG, X.; TANG, X. Deep learning face attributes in the wild. *In: IEEE. Proceedings of the IEEE international conference on computer vision*. Santiago, Chile: IEEE, 2015. p. 3730–3738.
- LONG J., S. E. . D. T. Fully convolutional networks for semantic segmentation. *In: IEEE. Proceedings of the IEEE conference on computer vision and pattern recognition*. Boston, MA: IEEE, 2015. p. 3431–3440.
- MACUÁCUA, J. C.; CENTENO, J. A. S.; AMISSE, C. Data mining approach for dry bean seeds classification. **Smart Agricultural Technology**, London, v. 5, n. 2, p. 1–12, 2023.
- MAIER, O.; MENZE, B. H.; GABLENTZ, J. von der; HÄNI, L.; HEINRICH, M. P.; LIEBRAND, M.; WINZECK, S.; BASIT, A.; BENTLEY, P.; CHEN, L. *et al.* Isles 2015-a public evaluation benchmark for ischemic stroke lesion segmentation from multispectral mri. **Medical image analysis**, London, v. 35, p. 250–269, 2017.
- MARTINEZ, A.; BENAVENTE, R. **The ar face database: Cvc technical report, 24**. Ohio, Columbus, 1998. Disponível em: <https://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>. Acesso em: 18 out. 2021.
- MEDEIROS, A.; PEIXOTO, S.; BARROS, C.; ALBUQUERQUE, V.; REBOUÇAS, P. P. Uma nova abordagem para a segmentação de pulmões utilizando o método de contorno ativo não paramétrico optimum path snakes em imagens de tomografia computadorizada. *In: Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*, 17., 2017, São Paulo. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2017. ISSN 2763-8952. DOI: <https://doi.org/10.5753/sbcas.2017.3699>.
- MEHMOOD, I.; ULLAH, A.; MUHAMMAD, K.; DENG, D.; MENG, W.; AL-TURJMAN, F.; SAJJAD, M.; ALBUQUERQUE, V. H. C. de. Efficient image recognition and retrieval on iot-assisted energy-constrained platforms from big data repositories. **IEEE Internet of Things Journal**, v. 6, n. 6, p. 1–1, 2019.
- MIRZAL, A. Statistical analysis of microarray data clustering using nmf, spectral clustering, kmeans, and gmm. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, IEEE, v. 19, n. 2, p. 1173–1192, 2020.
- MOON, A.; KIM, J.; ZHANG, J.; SON, S. W. Evaluating fidelity of lossy compression on spatiotemporal data from an iot enabled smart farm. **Computers and electronics in agriculture**, London, v. 154, p. 304–313, 2018.
- MUKHERJEE, D.; DAS, R.; MAJUMDAR, S.; GHOSH, S.; THEPADE, S.; BASU, A. Energy efficient face recognition in mobile-fog environment. **Procedia Computer Science**, London, v. 152, p. 274–281, 2019.
- NARGESIAN, F.; SAMULOWITZ, H.; KHURANA, U.; KHALIL, E. B.; TURAGA, D. S. Learning feature engineering for classification. **International Joint Conference on Artificial Intelligence**, v. 17, p. 2529–2535, 2017.

NASSRA, I.; CAPELLA, J. V. Data compression techniques in iot-enabled wireless body sensor networks: A systematic literature review and research trends for qos improvement. **Internet of Things**, London, p. 100806, 2023.

NGU, A. H.; GUTIERREZ, M.; METSIS, V.; NEPAL, S.; SHENG, Q. Z. Iot middleware: A survey on issues and enabling technologies. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 1, p. 1–20, 2016.

NITHILA, E. E.; KUMAR, S. Segmentation of lung from ct using various active contour models. **Biomedical Signal Processing and Control**, London, v. 47, p. 57–62, 2019.

NIXON, M.; AGUADO, A. Feature extraction and image processing for computer vision. Academic press, 2019.

PARISI, L.; RAVICHANDRAN, N. Evolutionary feature transformation to improve prognostic prediction of hepatitis. **Knowledge-Based Systems**, London, v. 200, n. 5, p. 7–20, 2020.

PEIXOTO, S.; PEDROSA, P.; KUMAR, A.; ALBUQUERQUE, V. Automatic classification of pulmonary diseases using a structural co-occurrence matrix. **Neural Computing and Applications**, Springer London, 09 2018. ISSN 0941-0643.

PULLAIAH, T.; MANJUNATHACHARI, K.; MALLESWARI, B. B δ -nis: Performance analysis of an efficient data compression technique for on-chip communication network. **Integration**, London, v. 89, p. 83–93, 2023.

QIAN, X.; WANG, J.; GUO, S.; LI, Q. An active contour model for medical image segmentation with application to brain ct image. **Medical physics**, Wiley Online Library, v. 40, n. 2, p. 1–10, 2013.

RAMALHO, G. L. B.; FERREIRA, D. S.; FILHO, P. P. R.; MEDEIROS, F. N. S. de. Rotation-invariant feature extraction using a structural co-occurrence matrix. **Measurement**, v. 94, p. 406 – 415, 2016. ISSN 0263-2241.

RAMALHO, G. L. B.; FILHO, P. P. R.; MEDEIROS, F. A. N. S. d.; CORTEZ, P. C. Lung disease detection using feature extraction and extreme learning machine. **Revista Brasileira de Engenharia Biomedica**, São Paulo, v. 30, p. 207 – 214, 2014. ISSN 1517-3151.

RANDHAWA, R. H.; HAMEED, A.; MIAN, A. N. Energy efficient cross-layer approach for object security of coap for iot devices. **Ad Hoc Networks**, London, v. 92, n. 3, p. 14–27, 2019.

REBOUÇAS, P. P.; BARROS, A. C. d. S.; ALMEIDA, J. S.; RODRIGUES, J. J. P. C.; ALBUQUERQUE, V. H. C. A new effective and powerful medical image segmentation algorithm based on optimum path snakes. **Applied Soft Computing**, London, v. 76, p. 649–670, 2019.

REBOUÇAS, P. P.; BARROS, A. C. d. S.; RAMALHO, G. L. B.; PEREIRA, C. R.; PAPA, J. P.; ALBUQUERQUE, V. H. C. de; TAVARES, J. M. R. S. Automated recognition of lung diseases in ct images based on the optimum-path forest classifier. **Neural Computing and Applications**, London, 2017. ISSN 1433-3058.

REBOUÇAS, P. P.; CORTEZ, P. C.; BARROS, A. C. da S.; ALBUQUERQUE, V. H. C.; TAVARES, J. M. R. Novel and powerful 3d adaptive crisp active contour method applied in the segmentation of ct lung images. **Medical image analysis**, London, v. 35, n. 2, p. 503–516, 2017.

REBOUÇAS, P. P.; REBOUÇAS, E. de S.; MARINHO, L. B.; SARMENTO, R. M.; TAVARES, J. M. R.; ALBUQUERQUE, V. H. C. de. Analysis of human tissue densities: A new approach to extract features from medical images. **Pattern Recognition Letters**, London, v. 94, p. 211 – 218, 2017. ISSN 0167-8655.

RODRIGUES, M. B.; NÓBREGA, R. V. M. D.; ALVES, S. S. A.; FILHO, P. P. R.; DUARTE, J. B. F.; SANGAIAH, A. K.; ALBUQUERQUE, V. H. C. D. Health of things algorithms for malignancy level classification of lung nodules. **IEEE Access**, IEEE, v. 6, p. 18592–18601, 2018.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986.

SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of pooling operations in convolutional architectures for object recognition. *In*: SPRINGER. **International conference on artificial neural networks**. Thessaloniki,GR, 2010. p. 92–101.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, v. 61, p. 85–117, 2015.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. *In*: IEEE. **Proceedings of the IEEE conference on computer vision and pattern recognition**. San Francisco, CA: IEEE, 2015. p. 815–823.

SHAH, D.; WANG, J.; HE, Q. P. Feature engineering in big data analytics for iot-enabled smart manufacturing—comparison between deep learning and statistical learning. **Computers & Chemical Engineering**, London, v. 141, p. 41–60, 2020.

SHAKIBAPOUR, E.; CUNHA, A.; ARESTA, G.; MENDONÇA, A. M.; CAMPILHO, A. An unsupervised metaheuristic search approach for segmentation and volume measurement of pulmonary nodules in lung ct scans. **Expert Systems with Applications**, London, v. 119, p. 415–428, 2019.

SHAKIR, H.; KHAN, T. M. R.; RASHEED, H. 3-d segmentation of lung nodules using hybrid level sets. **Computers in biology and medicine**, London, v. 96, p. 214–226, 2018.

SHEKHAWAT, S. S.; SHARMA, H.; KUMAR, S.; NAYYAR, A.; QURESHI, B. bssa: binary salp swarm algorithm with hybrid data transformation for feature selection. **Ieee Access**, IEEE, v. 9, p. 14867–14882, 2021.

SHEN, D.; WU, G.; SUK, H.-I. Deep learning in medical image analysis. **Annual review of biomedical engineering**, NIH Public Access, v. 19, p. 221, 2017.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 22, n. 8, p. 888–905, 2000.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. San Diego, v. 1, n. 2, p. 1–14, 2014.

SKOURT, B. A.; HASSANI, A. E.; MAJDA, A. Lung ct image segmentation using deep neural networks. **Procedia Computer Science**, London, v. 127, p. 109–113, 2018.

- SODHRO, A. H.; FORTINO, G.; PIRBHULAL, S.; LODRO, M. M.; SHAH, M. A. Energy-efficiency in wireless body sensor networks. **Networks of the Future: Architectures, Technologies, and Implementations**, CRC Press, p. 339, 2017.
- SODHRO, A. H.; LI, Y.; SHAH, M. A. Green and friendly media transmission algorithms for wireless body sensor networks. **Multimedia Tools and Applications**, v. 76, n. 4, p. 20001–20025, 2016.
- SODHRO, A. H.; LUO, Z.; SODHRO, G. H.; MUZAMAL, M.; RODRIGUES, J. J.; ALBUQUERQUE, V. H. C. de. Artificial intelligence based qos optimization for multimedia communication in iov systems. **Future Generation Computer Systems**, Londres, v. 95, n. 5, p. 667–680, 2019.
- SODHRO, A. H.; PIRBHULAL, S.; ALBUQUERQUE, V. H. C. de. Artificial intelligence-driven mechanism for edge computing-based industrial applications. **IEEE Transactions on Industrial Informatics**, v. 15, n. 7, p. 4235–4243, 2019.
- SODHRO, A. H.; PIRBHULAL, S.; QARAQE, M.; LOHANO, S.; SODHRO, G. H.; JUNEJO, N. U. R.; LUO, Z. Power control algorithms for media transmission in remote healthcare systems. **IEEE Access**, v. 6, n. 1, p. 42384–42393, 2018.
- SOUZA, J. W.; ALVES, S. S.; REBOUÇAS, E. d. S.; ALMEIDA, J. S.; FILHO, P. P. R. A new approach to diagnose parkinson’s disease using a structural cooccurrence matrix for a similarity analysis. **Computational intelligence and neuroscience**, Hindawi, v. 2018, n. 1, p. 2–9, 2018.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: <http://jmlr.org/papers/v15/srivastava14a.html>. Acesso em: 8 nov. 2022.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCHE, V.; RABINOVICH, A. Going deeper with convolutions. *In: IEEE. Proceedings of the IEEE conference on computer vision and pattern recognition*. Boston, MA: IEEE, 2015. p. 1–9.
- TURNER, C. R.; FUGGETTA, A.; LAVAZZA, L.; WOLF, A. L. A conceptual basis for feature engineering. **Journal of Systems and Software**, London, v. 49, n. 1, p. 3–15, 1999.
- ULLAH, A.; MUHAMMAD, K.; SER, J. D.; BAIK, S. W.; ALBUQUERQUE, V. H. C. de. Activity recognition using temporal optical flow convolutional features and multilayer lstm. **IEEE Transactions on Industrial Electronics**, v. 66, n. 12, p. 9692–9702, Dec 2019.
- WANG, C.; FRIMMEL, H.; SMEDBY, Ö. Level set based vessel segmentation accelerated with periodic monotonic speed function. **Medical Imaging 2011: Image Processing**, Bellingham, v. 7962, n. 8, p. 1–7, 2011.
- WANG, C.; FRIMMEL, H.; SMEDBY, Ö. Fast level-set based image segmentation using coherent propagation. **Medical physics**, Virgínia, v. 41, n. 7, p. 2–12, 2014.

WANG, M.; YANG, S.; WU, B. Hierarchical representation learning based spatio-temporal data redundancy reduction. **Neurocomputing**, London, v. 173, p. 298–305, 2016. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231215012722>. Acesso em: 13 ago. 2022.

XIAO, M.; WANG, D.; FU, Y.; LIU, K.; WU, M.; XIONG, H.; ZHOU, Y. Self-optimizing feature transformation. **arXiv preprint arXiv:2209.08044**, 2022. Disponível em: <https://arxiv.org/abs/2209.08044>. Acesso em: 4 jan. 2023.

XU, Q.; NING, L.; YUAN, T.; WU, H. Application of data mining combined with power data in assessment and prevention of regional atmospheric pollution. **Energy Reports**, London, v. 9, p. 3397–3405, 2023.

XU, X.; LIU, L.; ZHANG, X.; GUAN, W.; HU, R. Rethinking data collection for person re-identification: active redundancy reduction. **Pattern Recognition**, London, v. 113, p. 1–17, 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320321000145>. Acesso em: 15 dec. 2022.

YANG, X.; ZHANG, T.; XU, C.; YAN, S.; HOSSAIN, M. S.; GHONEIM, A. Deep relative attributes. **IEEE Transactions on Multimedia**, IEEE, v. 18, n. 9, p. 1832–1842, 2016.

YEMENI, Z.; WANG, H.; ISMAEL, W. M.; WANG, Y.; CHEN, Z. Reliable spatial and temporal data redundancy reduction approach for wsn. **Computer Networks**, London, v. 185, n. 4, p. 10–17, 2021.

ZENG, N.; ZHANG, H.; SONG, B.; LIU, W.; LI, Y.; DOBAIE, A. M. Facial expression recognition via learning deep sparse autoencoders. **Neurocomputing**, London, v. 273, p. 643–649, 2018.

ZHANG, H.; REN, S.; LI, X.; BAHARIN, H.; ALGHAMDI, A.; ALGHAMDI, O. Developing scalable management information system with big financial data using data mart and mining architecture. **Information Processing & Management**, London, v. 60, n. 3, p. 1–18, 2023.

ZHANG, S.; ZHAO, Y.; BAI, P. Object localization improved grabcut for lung parenchyma segmentation. **Procedia computer science**, London, v. 131, p. 1311–1317, 2018.

ZHUANG, H.; WANG, X.; BENDERSKY, M.; NAJORK, M. Feature transformation for neural ranking models. *In: ACM. Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. New York, NY: Association for Computing Machinery, 2020. p. 1649–1652.