UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

VALTER AUGUSTO DE FREITAS BARBOSA

SISTEMAS INTELIGENTES BASEADOS EM DEEP-WAVELET E REDES NEURAIS CONVOLUCIONAIS PARA APOIO AO DIAGNÓSTICO DE CÂNCER DE MAMA USANDO IMAGENS TERMOGRÁFICAS

Recife

2022

VALTER AUGUSTO DE FREITAS BARBOSA

SISTEMAS INTELIGENTES BASEADOS EM DEEP-WAVELET E REDES NEURAIS CONVOLUCIONAIS PARA APOIO AO DIAGNÓSTICO DE CÂNCER DE MAMA USANDO IMAGENS TERMOGRÁFICAS

Tese apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de doutor em Engenharia Mecânica.

Área de concentração: Energia.

Orientadora: Profª. Drª. Rita de Cássia Fernandes de Lima.
Orientador: Prof. Dr. Wellington Pinheiro dos Santos.

Recife

2022

Catalogação na Fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

B238s Barbosa, Valter Augusto de Freitas.

Sistemas inteligentes baseados em *deep-wavelet* e redes neurais convolucionais para apoio ao diagnóstico de câncer de mama usando imagens termográficas / Valter Augusto de Freitas Barbosa. - 2022.

155 folhas, il., gráfs., tabs.

Orientadora: Profa. Dra. Rita de Cássia Fernandes de Lima.

Orientador: Prof. Dr. Wellington Pinheiro dos Santos.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Mecânica, 2022.

Inclui Referências e Apêndices.

1. Engenharia Mecânica. 2. Câncer de mama. 3. Termografia. 4. Arquiteturas profundas híbridas. 5. *Deep-wavelet neural networks*. 6. Redes neurais convolucionais. 7. Diagnóstico. I. Lima, Rita de Cássia Fernandes de (Orientadora). II. Santos, Wellington Pinheiro dos (Orientador). III. Título.

UFPE

621 CDD (22. ed.) BCTG/2022-146

VALTER AUGUSTO DE FREITAS BARBOSA

SISTEMAS INTELIGENTES BASEADOS EM DEEP-WAVELET E REDES NEURAIS CONVOLUCIONAIS PARA APOIO AO DIAGNÓSTICO DE CÂNCER DE MAMA USANDO IMAGENS TERMOGRÁFICAS

Tese apresentada ao Programa de Pós-Graduação em Engenharia Mecânica do Departamento de Engenharia Mecânica, Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco como parte dos requisitos parciais para obtenção do título de doutor em Engenharia Mecânica.

Aprovado em: 10/02/2022

BANCA EXAMINADORA

_____

Profª. Drª. Rita de Cássia Fernandes de Lima (Orientadora)
Universidade Federal de Pernambuco

_____

Prof. Dr. Fábio Santana Magnani (Examinador Interno)
Universidade Federal de Pernambuco

_____

Prof. Dr. Marcus Costa de Araújo (Examinador Externo)
Universidade Federal de Pernambuco

_____

Prof. Dr. Ricardo Emmanuel de Souza (Examinador Externo)
Universidade Federal de Pernambuco

_____

Prof. Dr. Sidney Marlon Lopes de Lima (Examinador Externo)
Universidade Federal de Pernambuco

_____

Profª. Drª. Giselle Machado Magalhães Moreno (Examinadora Externa)
Universidade de São Paulo

Este trabalho é dedicado a todos que acreditam na ciência.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar saúde e força para que eu chegasse até aqui. E especialmente, por ter me mantido e aos meus próximos com saúde física e mental durante a pandemia de Covid-19.

Agradeço ao Professor Wellington por sua notável contribuição a este trabalho, por sempre acreditar em mim e por não medir esforços ao me fornecer oportunidades para que eu pudesse evoluir como pesquisador e docente. Sou muito grato à Professora Rita por ter me aceitado como seu aluno, pelos seus conselhos e seu olhar crítico, responsável por melhorar a qualidade da pesquisa e desta tese. Nunca esquecerei o apoio inestimável de ambos, nos momentos em que estive me preparando para os concursos da carreira docente. Sem este apoio eu não seria hoje professor da Universidade Federal Rural de Pernambuco.

Às minhas colegas de pesquisa, Clarice, Juliana e Maíra, por compartilharmos inúmeros desafios que enfrentamos e pelos bons momentos passados durante as reuniões remotas.

À minha esposa Amanda, minha parceira de vida, pelo seu apoio incondicional e por estar presente em todos os momentos, desde os mais difíceis quanto os mais alegres.

À minha família, pela paciência e compreensão para comigo nessa árdua jornada que é a pós-graduação.

Por fim, gostaria de agradecer à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

RESUMO

O câncer de mama é o tipo de câncer mais frequente e mortal entre as mulheres. Contudo, quanto mais cedo o câncer for diagnosticado melhores são as chances de recuperação da paciente. Atualmente, o exame mais bem aceito para a detecção do câncer de mama em pacientes assintomáticas é a mamografia. Porém, a mamografia é um exame que possui uma alta taxa de falso positivo. Além de ser um exame desconfortável, e que expõe a paciente a radiações ionizantes. Neste contexto, uma técnica emergente é a termografia de mama, a qual foi aprovada em 1982 pelo *Food and Drugs Administration* (FDA) como técnica auxiliar à mamografia. A termografia é uma técnica de menor custo comparada à mamografia, livre de radiações ionizantes e que não causa nenhum desconforto à paciente. Essa técnica é capaz de gerar uma imagem que apresenta medidas da distribuição de temperatura superficial da mama. É possível detectar lesões cancerígenas em imagens de termografia ao identificar perturbações no padrão de temperatura da mama, pois o crescimento cancerígeno está associado a eventos de maior produção de calor da região afetada, à neoangiogênese (produção de novos vasos sanguíneos) e ao aumento do fluxo sanguíneo. Por outro lado, a análise das imagens de termografia não é uma tarefa fácil. Sendo assim, o uso de técnicas da inteligência artificial para a análise das imagens pode desempenhar um papel relevante para a ampla utilização da termografia na detecção do câncer de mama. Este trabalho propõe o desenvolvimento de uma ferramenta baseada em técnicas da inteligência artificial para a detecção e classificação de lesões mamárias em imagens de termografia de mama. Além disso, neste trabalho é apresentada a formalização matemática de uma técnica da aprendizagem profunda para extração de atributos de imagens baseado na decomposição *wavelet*, chamada de *Deep-Wavelet Neural Network* (DWNN). Este método foi aplicado ao problema de classificação de imagens de termografia e seus resultados foram comparados com os resultados de seis redes neurais convolucionais do estado da arte. Os experimentos foram divididos de acordo com duas abordagens. Na primeira abordagem o objetivo foi detectar lesões mamárias entre imagens com e sem lesão. Na segunda abordagem o objetivo foi de classificar lesões entre as imagens de pacientes com cisto, lesão benigna e lesão maligna. Os melhores resultados foram obtidos ao utilizar a DWNN com seis camadas, tendo seus atributos selecionados pela *Random Forest* e classificados com a Máquina de Vetor de Suporte com *kernel* polinomial linear. Para a primeira abordagem a ferramenta atingiu: 99% de acurácia e 0,98 para o índice

kappa. Enquanto que para a segunda abordagem o método conseguiu 97,3% de acurácia e 0,96 para o índice kappa.

Palavras-chave: câncer de mama; termografia; arquiteturas profundas híbridas; *deep-wavelet neural networks*; redes neurais convolucionais; diagnóstico.

ABSTRACT

Breast cancer is the most common and deadly type of cancer among women. However, the earlier the cancer is diagnosed, the better the patient's chances of recovery. Currently, the most widely accepted test for detecting breast cancer in asymptomatic patients is mammography. However, mammography is a test that has a high rate of false positives. Moreover, it is an uncomfortable test, and also it exposes the patient to ionizing radiation. In this context, an emerging technique is breast thermography, which was approved in 1982 by the Food and Drugs Administration (FDA) as an auxiliary technique to mammography. Thermography is a lower cost technique compared to mammography, free of ionizing radiation and does not cause any discomfort to the patient. This technique is capable of generating an image that represents the breast surface temperature distribution measures. It is possible to detect cancerous lesions in thermography images by identifying disturbances in the breast temperature pattern, as cancerous growth is associated with events of greater heat production in the affected region, the neoangiogenesis (production of new blood vessels) and increased flow blood. On the other hand, analyzing thermography images is not an easy task. Thus, the use of artificial intelligence techniques for image analysis can play a relevant role in the wide use of thermography in the detection of breast cancer. This work proposes the development of a tool based on artificial intelligence techniques for the detection and classification of breast lesions in breast thermography images. In addition, this work presents the mathematical formalization of a deep learning technique for image attribute extraction based on wavelet decomposition, called the Deep-Wavelet Neural Network (DWNN). This method was applied to the thermography image classification problem and its results were compared with the results of six state-of-the-art convolutional neural networks. The experiments were divided according to two approaches. In the first approach, the objective is to detect breast lesions between images with and without lesions. In the second approach, the objective is to classify lesions among the images from patients with cysts, benign lesions and malignant lesions. The best results were obtained using the DWNN with six layers, having its attributes selected by Random Forest and classified with the Support Vector Machine with linear polynomial kernel. For the first approach, the tool reached 99% of accuracy and 0.98 for kappa index. While for the second approach, the method achieved 97.3% of accuracy and 0.96 for kappa index.

Keywords: breast cancer; thermography; hybrid deep architectures; deep-wavelet neural networks; convolutional neural networks; diagnosis.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Segundo a Organização Mundial de Saúde (OMS), o câncer de mama é o tipo de câncer mais frequente entre as mulheres, afetando 2,1 milhões de mulheres a cada ano no mundo (OMS, 2019). Além disso, o câncer de mama é responsável pelo maior número de mortes relacionadas a cânceres entre as mulheres. Em 2018, estima-se que 627.000 mulheres morreram devido ao câncer de mama, isto representa 15% de todas as mortes de mulheres devido a um câncer (OMS, 2019). No Brasil, o Instituto Nacional de Câncer José Alencar Gomes da Silva (INCA) estimou que 66.280 novos casos da doença foram diagnosticados em 2020 (INCA, 2021).

A distribuição de novos casos no Brasil é heterogênea, onde as regiões do país que possuem as maiores taxas de incidência são as Regiões Sul e Sudeste, com 59,13 e 56,58 novos casos por 100 mil habitantes respectivamente, o que representa 70% dos novos casos de todo o país. Enquanto que as menores taxas de incidência ocorrem nas Regiões Norte e Nordeste, com 24,33 e 38,84 respectivamente (representando aproximadamente 23%). Numa situação intermediária encontra-se a região Centro-Oeste com 51,29 novos casos por 100 mil habitantes (representando 7% dos novos casos) (INCA, 2018b).

Os fatores de risco que podem desencadear o câncer de mama podem ser classificados em três categorias: fatores ambientais e comportamentais; fatores da história reprodutiva e hormonal; e fatores genéticos e hereditários. Na Tabela 1 são mostrados alguns fatores de riscos de acordo com essa classificação para o câncer de mama. Contudo, possuir algum desses fatores de riscos não significa que a mulher vá desenvolver a doença algum dia (INCA, 2018a).

Alguns hábitos como praticar atividade física, manter o peso corporal adequado, evitar o consumo de bebidas alcoólicas e amamentar podem diminuir em cerda de 30% os casos de câncer de mama (INCA, 2018a).

A melhor estratégia, atualmente, para diminuir a morbidade e a mortalidade da doença é a detecção precoce (WALKER; KACZOR, 2012), em outras palavras, descobrir o tumor em sua fase inicial aumentam as chances de a paciente se curar da doença.

Há duas estratégias para a detecção precoce do câncer de mama: o diagnóstico

Tabela 1 – Fatores de risco para o câncer de mama.

Fatores ambientais e comportamentais	**Fatores da história reprodutiva e hormonal**	**Fatores genéticos e hereditários**
Obesidade e sobrepeso após a menopausa;	Primeira menstruação antes de 12 anos;	História familiar de câncer de ovário;
Sedentarismo e inatividade física;	Não ter tido filhos;	Casos de câncer de mama na família, principalmente antes dos 50 anos;
Consumo de bebida alcoólica;	Primeira gravidez após os 30 anos;	História familiar de câncer de mama em homens;
Exposição frequente a radiações ionizantes.	Parar de menstruar (menopausa) após os 55 anos;	Alteração genética, especialmente nos genes BRCA1 e BRCA2.
	Uso de contraceptivos hormonais (estrogênio-progesterona).	
	Ter feito reposição hormonal pós-menopausa, principalmente por mais de cinco anos.	

Fonte: INCA (2018a).

precoce e o rastreamento. O diagnóstico precoce consiste em identificar, o mais cedo possível, o câncer de mama em indivíduos sintomáticos, enquanto que o rastreamento é a identificação do câncer de mama em indivíduos assintomáticos (INCA, 2015; OMS, 2019).

A estratégia do diagnóstico precoce do câncer de mama é formada pelo tripé: população alerta dos sinais e sintomas do câncer; profissionais de saúde capacitados para a avaliação; e sistemas e serviços de saúde dedicados a confirmação do diagnóstico. Como estratégia, o rastreamento adota a execução de testes relativamente simples em pessoas sadias, visando a identificação do câncer na sua fase pré-clínica (assintomática) (INCA, 2015).

O exame mais bem aceito utilizado no rastreamento da população em geral é a mamografia. No Brasil, a realização bienal da mamografia é indicada para mulheres de 50 a 69 anos de idade (INCA, 2015). Porém, a sensibilidade da mamografia depende da composição do tecido mamário, onde a técnica possui bons resultados para pacientes com tecido adiposo, mas resultados baixos para pacientes com tecido glandular (EULER-CHELPIN et al., 2019). Além disso, a mamografia possui uma alta taxa de falsos positivos (EULER-

CHELPIN et al., 2019) e submete a paciente a radiações ionizantes. O que oferece riscos as pacientes, sobretudo as mais jovens, que possuem o tecido mamário essencialmente glandular, o qual é opaco para radiações ionizantes. Sem contar que a realização do exame se dá através da compressão da mama, causando desconforto à paciente. Ainda assim, em termos de precisão, custo, acesso e riscos, a mamografia é o método de rastreamento melhor definido do que qualquer outro método (WALKER; KACZOR, 2012).

Além da mamografia, outras técnicas como ultrassonografia, ressonância magnética nuclear, cintilografia, termografia e tomografia por impedância elétrica podem ser utilizadas no rastreamento como ferramentas complementares no diagnóstico do câncer de mama (WALKER; KACZOR, 2012; CHEREPENIN et al., 2001).

Além dessas, a termografia de mama é uma técnica alternativa que vem ganhando proeminência nas últimas décadas como um método complementar à mamografia no rastreamento do câncer de mama (SANTANA et al., 2018). Em 1982, a termografia de mama foi aprovada como modalidade de imagem adjunta à mamografia pela *Food and Drug Administration* (FDA) (SINGH; SINGH, 2020; ARORA et al., 2008). Essa técnica é caracterizada por ser não-invasiva, indolor, sem contato físico e de baixo custo com relação à mamografia, e à ressonância magnética (BORCHARTT et al., 2013). A termografia de mama pode ser utilizada em todas as mulheres de todas as idades, como grávidas, lactantes, mulheres com implantes, mulheres com mama densa ou apresentando fibrocistos, mulheres submetidas à terapia de reposição hormonal, e tanto para mulheres na pré-menopausa quanto na pós-menopausa (ETEHADTAVAKOL; NG, 2013).

A termografia de mama é baseada na obtenção de uma imagem que representa a distribuição de temperatura da pele da paciente, isto é, a distribuição de temperatura superficial da mama. Para um dado tecido sadio tem-se um determinado padrão de distribuição de temperatura que é função da sua atividade metabólica. Nesse contexto, pode-se entender a atividade metabólica como um processo de geração de calor. Sendo assim, quando qualquer distúrbio fisiológico altera a atividade metabólica de um tecido a sua distribuição de temperatura também será alterada. Assim tal distúrbio poderá ser identificado através de uma imagem de termografia.

A evolução de um tumor está associada à neoangiogênese (produção de novos vasos sanguíneos) e ao aumento do fluxo sanguíneo na região afetada. Esta alteração

do tecido resulta em um acréscimo da temperatura local da pele em 1 a 2 ℃ (MEIRA et al., 2014), sendo assim, a evolução de um tumor é perceptível à técnica de imagens termográficas.

Um dos primeiros registros da utilização do uso da temperatura da mama como forma de detectar tumores data da década de 50 (LAWSON, 1956). Nos anos que se seguiram, a técnica mostrava possuir potencial, contudo, os recursos tecnológicos da época não eram suficientes para obter bons resultados na detecção de lesões mamárias (WALKER; KACZOR, 2012) fazendo com que a técnica ficasse desacreditada. O uso de imagens de termografia veio aumentar apenas quando câmeras termográficas mais precisas e sensíveis foram desenvolvidas (ETEHADTAVAKOL; NG, 2013; MOGHBEL; MASHOHOR, 2013).

De fato, a termografia de mama é um teste funcional (fisiológico), o que permite a identificação do tumor ainda em fases iniciais durante o seu crescimento. Como frequentemente mudanças fisiológicas precedem mudanças anatômicas, a termografia de mama consegue identificar o tumor ainda nos estágios iniciais antes mesmo da mamografia (a qual se trata de um exame anatômico) (SCHAEFER; NAKASHIMA; ZAVISEK, 2008; ETEHADTAVAKOL; NG, 2013). Através da análise da distribuição de temperatura e dos vasos sanguíneos da mama, alguns autores citam que sinais de um possível câncer ou uma expansão de células pré-cancerosas podem ser reconhecidas até mesmo 10 anos mais cedo do que outras técnicas (BORCHARTT et al., 2013; ETEHADTAVAKOL; NG, 2013).

Apesar de ser uma técnica promissora, a análise de imagens termográficas não é uma tarefa simples. Basicamente, comparam-se duas imagens contralaterais. Pequenas assimetrias podem demonstrar uma região de anormalidade quando as imagens são quase simétricas. Contrariamente, esses pequenos desequilíbrios podem não ser fáceis de identificar (ETEHADTAVAKOL; NG, 2013). A análise torna-se ainda mais difícil quando o tumor é profundo, pois sua interferência causada na temperatura da pele da mama não é de forma pontual, intensa e de fácil visualização, mas sim de uma forma distribuída e de pouca intensidade. Portanto, o desenvolvimento de um método automático para eliminar fatores humanos é importante para se obter melhores resultados com a termografia (ETEHADTAVAKOL; NG, 2013).

Nesse contexto, sistemas de diagnóstico assistido por computador (também conhecido pela sigla CAD de *Computer-Aided Diagnosis Systems*) aplicados à análise de

imagens de termografia de mama podem assistir profissionais médicos. Pois, a termografia de mama aliada a um sistema automático de análise podem servir como um sistema de triagem para a detecção de lesões mamárias, especialmente em regiões onde há carência de profissionais especialistas, como em regiões de poucos recursos e de difícil acesso. Além disso, seres humanos estão sujeitos a efeitos do estresse, fadiga e dias difíceis, o que pode comprometer a análise das imagens. Nesse sentido, os sistemas automáticos de análise de imagens podem servir como uma ferramenta de apoio ao diagnóstico, atuando como uma ferramenta de segunda opinião.

As redes neurais convolucionais (*Convolutional Neural Networks* - CNNs) são modelos da aprendizagem profunda caracterizadas por ter muitas camadas e por serem capazes de realizar convoluções. As CNNs têm grandes aplicações em problemas de aprendizado de máquina, como classificação de vídeo (KARPATHY et al., 2014; TRAN et al., 2019), segmentação de imagem (LONG; SHELHAMER; DARRELL, 2015; RONNEBERGER; FISCHER; BROX, 2015; YANG et al., 2020), reconhecimento de voz (LIANG et al., 2017) e reconhecimento de modulação (ZHOU; LIU; GRAVELLE, 2020). Contudo, o maior domínio de aplicação das CNNs é a classificação de imagem (O'SHEA; NASH, 2015; SZEGEDY et al., 2015; SZEGEDY et al., 2016; ALBAWI; MOHAMMED; AL-ZAWI, 2017; CHOLLET, 2017). Sendo assim, vários estudos vêm aplicando modelos de CNNs à classificação de imagens de termografia mamária para a detecção do câncer de mama (BAFFA; LATTARI, 2018; ROSLIDAR et al., 2019; CHAVES et al., 2020; EKICI; JAWZAL, 2020; MISHRA et al., 2020; MAMBOU et al., 2018).

Essa tese de doutorado possui como contribuição a investigação do uso de técnicas da aprendizagem de máquina para a extração de atributos de imagens de termografia de mama. Além disso, é proposto um método para a extração de atributos chamado *Deep-Wavelet Neural Network* (DWNN). As imagens de termografia de mama também são avaliadas por seis redes neurais convolucionais do estado da artes pré-treinadas no conjunto de imagens ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). O conjunto de imagens de termografia utilizado foi obtido no Hospital das Clínicas da Universidade Federal de Pernambuco (HC-UFPE) e possui o diferencial de possuir imagens de pacientes com cisto e com lesão benigna, além da lesão maligna. Diferentes de outras base, como a *Database for Mastology Research with Infrared Image* - (DMR) (SILVA et al., 2014), as quais muitos estudos são baseados (ver o Capítulo 3), que possuem apenas imagens de

pacientes sadias e com lesão maligna. Desta forma, neste trabalho os experimentos são realizados com imagens rotuladas como pacientes com cisto, com lesão benigna, com lesão maligna e sem lesão. O sistema desenvolvido é baseado em máquinas de aprendizados conexionistas otimizadas por um algoritmo de árvores de decisão. O trabalho visa identificar com maior acurácia, sensibilidade, especificidade, índice kappa e precisão as imagens de termografia, servindo assim como dispositivo para o apoio à detecção precoce do câncer de mama. O sistema contém duas abordagens para a identificação de lesões, uma para diferenciar imagens sem lesão daquelas que possuem qualquer tipo de lesão, e outra para classificar imagens com lesões entre: com cisto, com lesão benigna e com lesão maligna.

1.1 Contexto e motivação

A termografia de mama demonstra-se como um potencial método auxiliar no rastreamento do câncer de mama, justamente por ser capaz de identificar o câncer em suas fases iniciais antes mesmo que outros métodos já bem consolidados na área como a mamografia (SCHAEFER; NAKASHIMA; ZAVISEK, 2008; ETEHADTAVAKOL; NG, 2013). Além disso, a termografia é um método não-invasivo, sem contato e livre de radiações ionizantes. Contudo, a correta interpretação das imagens de infravermelho ainda continua sendo um desafio. Uma das alternativas para superar tal desafio vem através do desenvolvimento de métodos automáticos de análise das imagens utilizando técnicas de processamento digital de imagens e da inteligência artificial. Dessa forma, o trabalho aqui proposto visa contribuir na área de termografia de mama através do desenvolvimento de formas automáticas para classificar lesões mamárias em imagens de infravermelho.

1.2 Objetivos

Este projeto tem como objetivo principal desenvolver um sistema inteligente, baseado em algoritmos de aprendizagem profunda, para apoio ao diagnóstico não-invasivo do câncer de mama por meio da análise automatizada de imagens termográficas de mama. Além disso, neste trabalho, é feita a formalização matemática da *Deep-Wavelet Neural Network*, um método da aprendizagem profunda, para a extração de atributos, baseado na decomposição *wavelet*. Método este que é aplicado ao problema da classificação de imagens de termografia.

Este projeto tem os seguintes objetivos específicos:

- Escrever a formalização matemática da *Deep-Wavelet Neural Network*;
- Aplicar a *Deep-Wavelet Neural Network* e redes neurais convolucionais para a extração de atributos das imagens de termografia.
- Realizar o balanceamento das bases obtidas pelos extratores de atributos no mapa das características;
- Identificar os atributos mais relevantes para cada base através da seleção de atributos;
- Construir uma máquina de aprendizado conexionista para detecção de lesão em imagens termográficas de mama;
- Construir uma máquina de aprendizado conexionista para a classificação de lesões em imagens termográficas de mama.

1.3 Organização do trabalho

Este trabalho está organizado da seguinte forma. No Capítulo 2 são apresentados os conceitos teóricos necessários ao entendimento do trabalho, como a fundamentação sobre a termografia, e os conceitos de inteligência artificial como as redes neurais convolucionais para a classificação de imagens. No Capítulo 3 são apresentados trabalhos relacionados no contexto do uso da termografia de mama para a detecção do câncer de mama. O Capítulo 4 apresenta a metodologia do trabalho, como também, a formalização matemática da *Deep-Wavelet Neural Network*, um dos objetivos deste trabalho. Os resultados experimentais são mostrados e discutidos no Capítulo 5. Por fim, o trabalho é finalizado no Capítulo 6 com as conclusões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo é destinado à revisão de conceitos teóricos importantes e necessários para a compreensão desta pesquisa.

2.1 Termografia de mama

Todo corpo com temperatura acima do zero absoluto[1] emite ondas eletromagnéticas, também chamadas de radiação térmica (HALLIDAY; RESNICK; WALKER, 2016). A maior parte das ondas eletromagnéticas emitidas por corpos ao nosso redor possuem frequência na faixa do infravermelho. Contudo em alguns casos essa radiação é emitida no espectro visível, como é o caso de metais aquecidos a altas temperaturas que passam a ser incandescentes emitindo uma luz avermelhada. Um exemplo prático disso é a lâmpada incandescente que utiliza um filamento de tungstênio aquecido a altas temperaturas para a geração de luz visível.

A taxa de emissão de energia através da radiação térmica, por unidade de tempo, é dada pela Equação 2.1, onde $\sigma = 5,6704 \cdot 10^{-8}\ W/m^2 \cdot K^4$ é a constante de Stefan-Bolztmann, ϵ é a emissividade, uma propriedade da superfície do objeto, A e T são, respectivamente, a área superficial e a temperatura em Kelvin do objeto (HALLIDAY; RESNICK; WALKER, 2016).

$$P_{rad} = \sigma \epsilon A T^4 \tag{2.1}$$

Contudo, da mesma forma que um objeto emite radiação térmica, ele também absorve radiações térmicas de outros corpos que estão ao seu redor. Dessa forma, a taxa líquida, entre a energia absorvida e emitida por radiação térmica é dada pela Equação 2.2, onde T_{amb} é a temperatura (em Kelvin) do ambiente em que o objeto está inserido, onde supõe-se que esta é uniforme (HALLIDAY; RESNICK; WALKER, 2016).

$$P_{liq} = \sigma \epsilon A (T_{amb}^4 - T^4) \tag{2.2}$$

[1] Limite inferior da temperatura dado pelo valor 0 na escala Kelvin (0 K) que equivale a - 273,15 °C.

A pele humana emite radiação infravermelha, essencialmente, com comprimentos de onda dentro da faixa de 2-20 μm e com um pico médio entre 9-10 μm (ETEHADTAVAKOL; NG, 2013). Uma câmera de termografia é feita com sensores sensíveis à radiação eletromagnética no espectro do infravermelho, e não na faixa da luz visível, como as câmeras fotográficas comuns. Dessa forma, a radiação térmica (infravermelha) emitida por um ponto da pele pode ser convertido diretamente em um valor de temperatura que representa esse ponto e então mapeado em um *pixel* de uma imagem digital (BORCHARTT et al., 2013). Logo, uma imagem termográfica é um mapeamento da distribuição de temperatura de uma dada região. Tal imagem pode ser apresentada tanto em níveis de cinza quanto colorida artificialmente utilizando alguma paleta de falsa cor.

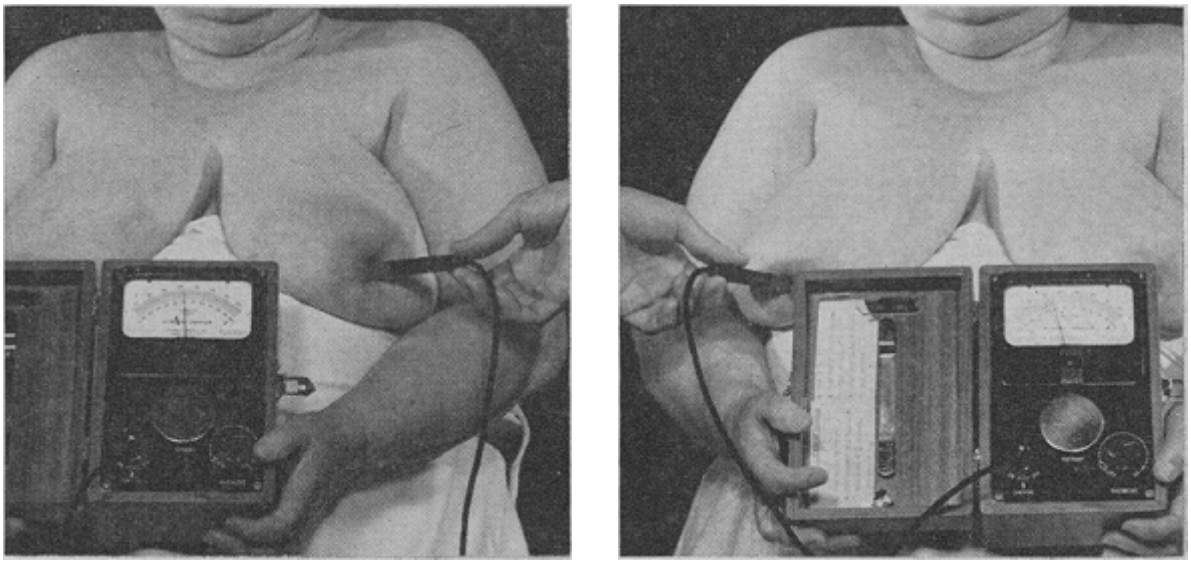
Um dos primeiros trabalhos a relacionar a temperatura da superfície da mama e a presença de um tumor foi realizado por LAWSON em 1956. Seu estudo foi baseado na medição da temperatura da pele diretamente através de um termopar. Ao trabalhar com um grupo de 26 pacientes com câncer de mama ele constatou haver um aumento médio de temperatura de 2,27 °F (equivalente a 1,26 °C) da área do tumor ou na aréola ipsilateral (como mostrado na Figura 1). No mesmo artigo, ele faz referência sobre o trabalho realizado por Massopust[2] em imagens de infravermelho dos vasos sanguíneos superficiais, mas ele termina seu comentário argumentando sobre as limitações técnicas da época que inviabilizavam a técnica. De fato, problemas iniciais como a baixa sensibilidade dos detectores e com limitações técnicas representavam uma enorme fonte de erro para a termografia limitando e retardando a sua aceitação até a década de 90 (MEIRA et al., 2014; ETEHADTAVAKOL; NG, 2013), quando a tecnologia de sensores de infravermelho deixou de ser de uso restrito para fins militares. Nessa época surgiu no mercado um novo tipo de câmera que deu início à chamada segunda geração de câmeras infravermelhas, que corrigiu alguns problemas de câmeras anteriores como flutuações térmicas devido ao calor interno do equipamento durante o uso (conhecidas como *thermal drift*), baixa sensibilidade e tempo de aquisição de imagens (HEAD et al., 1996; MOGHBEL; MASHOHOR, 2013).

Hoje em dia, possuímos câmeras capazes de fornecer imagens termográficas digitais. O que é uma grande vantagem por permitir a manipulação e aplicação de técnicas computacionais e de processamento digital de imagens na avaliação dessas imagens, como as redes neurais artificiais (WALKER; KACZOR, 2012). Nas Figuras 2, 3, 4 e 5 são

[2] Também considerado como um dos pioneiros dos estudos da termografia de mama (WILLIAMS; WILLIAMS, 2002).

Figura 1 – Variação de temperatura nas mamas de uma paciente com carcinoma unilateral

(a) Mama esquerda com carcinoma (b) Mama direita sem anomalia



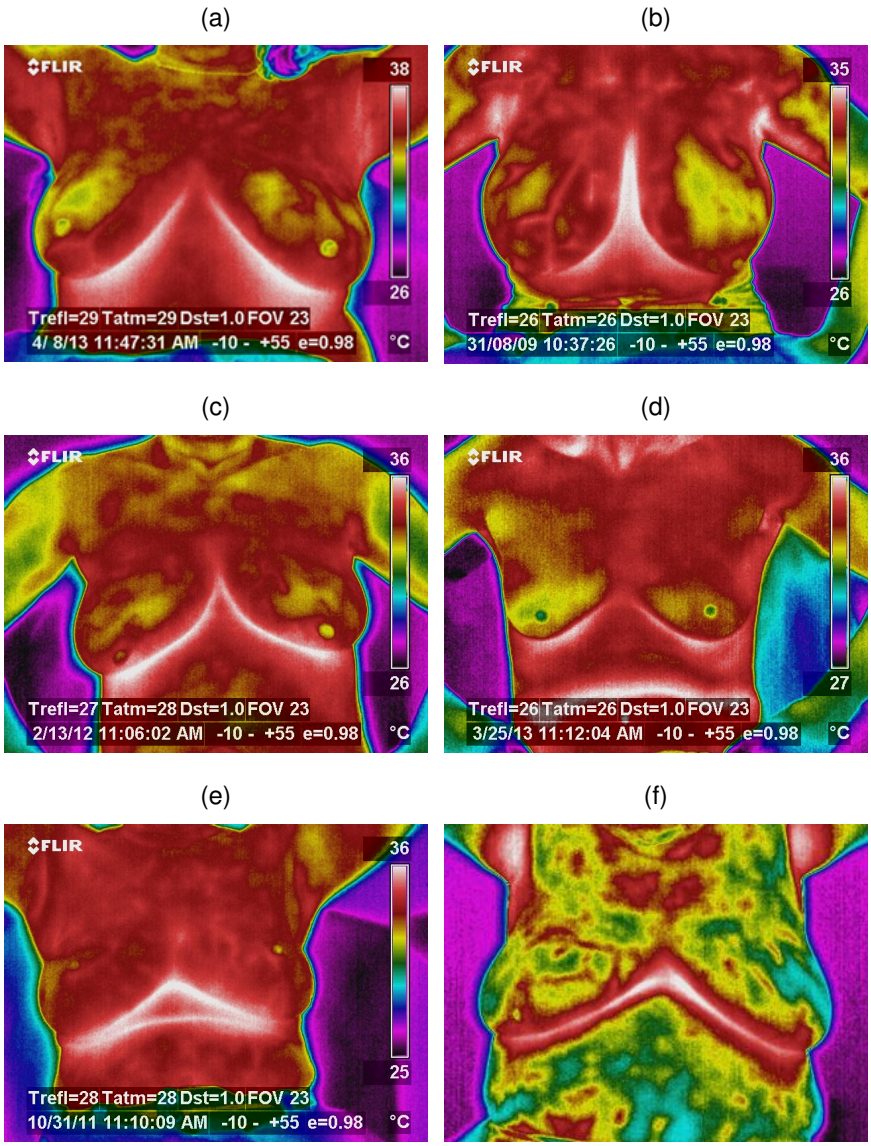
Fonte: Lawson (1956).

mostradas diferentes imagens de termografia para pacientes sem lesão, com cistos, com lesão benigna e com lesão maligna, respectivamente.

Na área médica, além da sua aplicação na detecção de tumores mamários, a termografia também pode ser usada na ortopedia, odontologia, cardiologia, endocrinologia, medicina forense, hemodinâmica, obstetrícia, fisioterapia e ergonomia (MEIRA et al., 2014).

Termogramas são sensíveis a mudanças de temperatura, umidade e ventilação do ambiente. Dessa forma, a sua aquisição deve ser feita em condições controladas (BORCHARTT et al., 2013). As imagens utilizadas neste trabalho foram obtidas através do protocolo desenvolvido pela equipe do Departamento de Engenharia Mecânica da Universidade Federal de Pernambuco (UFPE) para a aquisição de imagens termográficas de mama. Tal protocolo foi proposto por Oliveira (2012) e também pode ser encontrado em Bezerra et al. (2013) e Santana et al. (2018). A construção dessa base e outros trabalhos fazem parte do projeto de pesquisa intitulado “Análise da viabilidade do uso de câmera termográfica como ferramenta auxiliar no diagnóstico de câncer de mama em hospital público localizado em clima tropical”, que foi aprovado pelo Comitê de Ética da Universidade Federal de Pernambuco (UFPE) – Brasil, e registrado no Ministério de Saúde sob o número CEP/CCS/UFPE Nº279/05, e que se encontra em andamento desde 2005. As imagens foram obtidas no Hospital das Clínicas da UFPE. A Figura 6 mostra o fluxograma do protocolo, o qual se resume nas etapas de adequação da sala, preparação da paciente

Figura 2 – Imagens de termografia de mama de pacientes sem lesão.

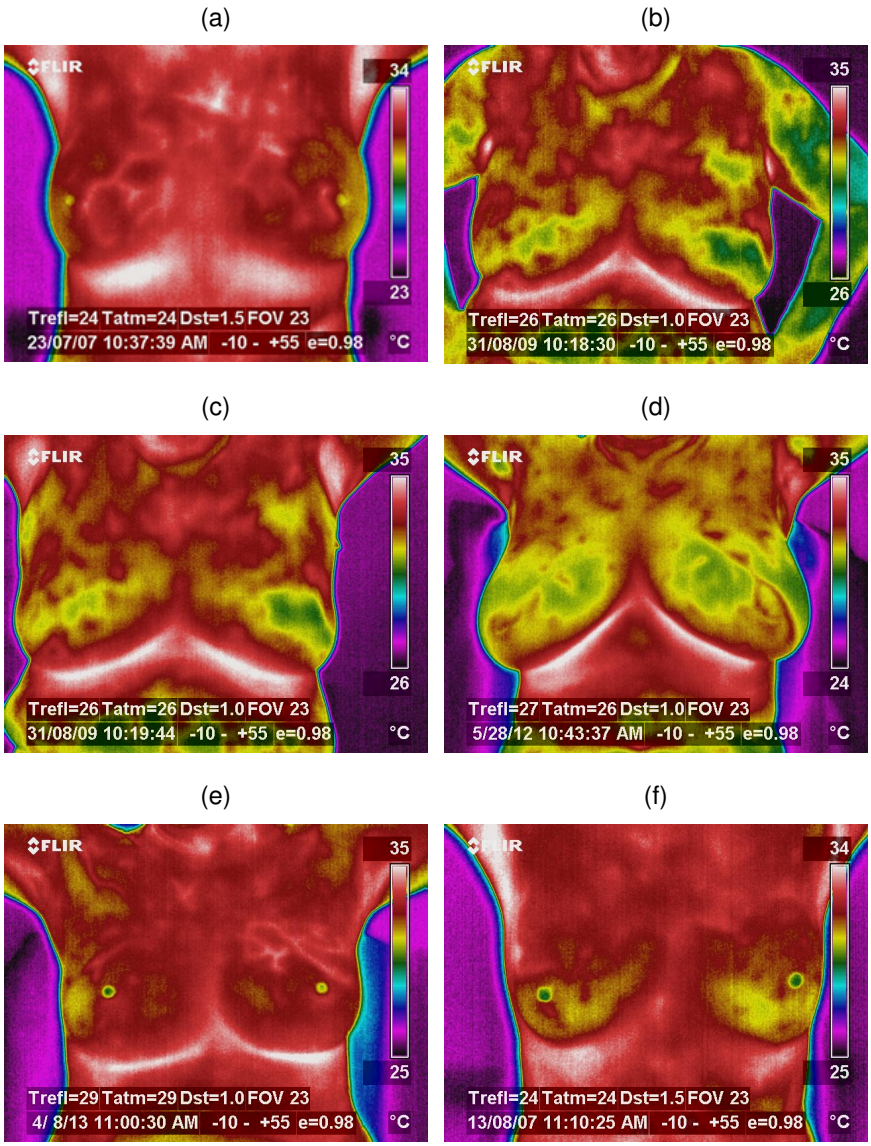


Fonte: O Autor (2022).

e a aquisição das imagens. Na etapa de adequação da sala, o ambiente é climatizado de modo a evitar ruídos externos nas imagens. Além disso, aferições da temperatura e umidade relativa da sala é feita para configurar a câmera de infravermelho. Em seguida, as pacientes são preparadas para o procedimento de modo que sua temperatura corporal esteja estabilizada. Por fim, a imagens termográficas são obtidas.

De modo a padronizar as imagens termográficas, a aquisição é feita através de um aparato para acomodar a paciente e a câmera termográfica, como é mostrado na Figura 7. O aparato é formado por uma cadeira giratória, para qual há um suporte superior onde a paciente pode posicionar seus braços. Por fim, a câmera é posta, com a ajuda de um tripé, sobre um carrinho sobre trilhos. A função dos trilhos é auxiliar no ajuste da distância entre a

Figura 3 – Imagens de termografia de mama de pacientes com cisto.

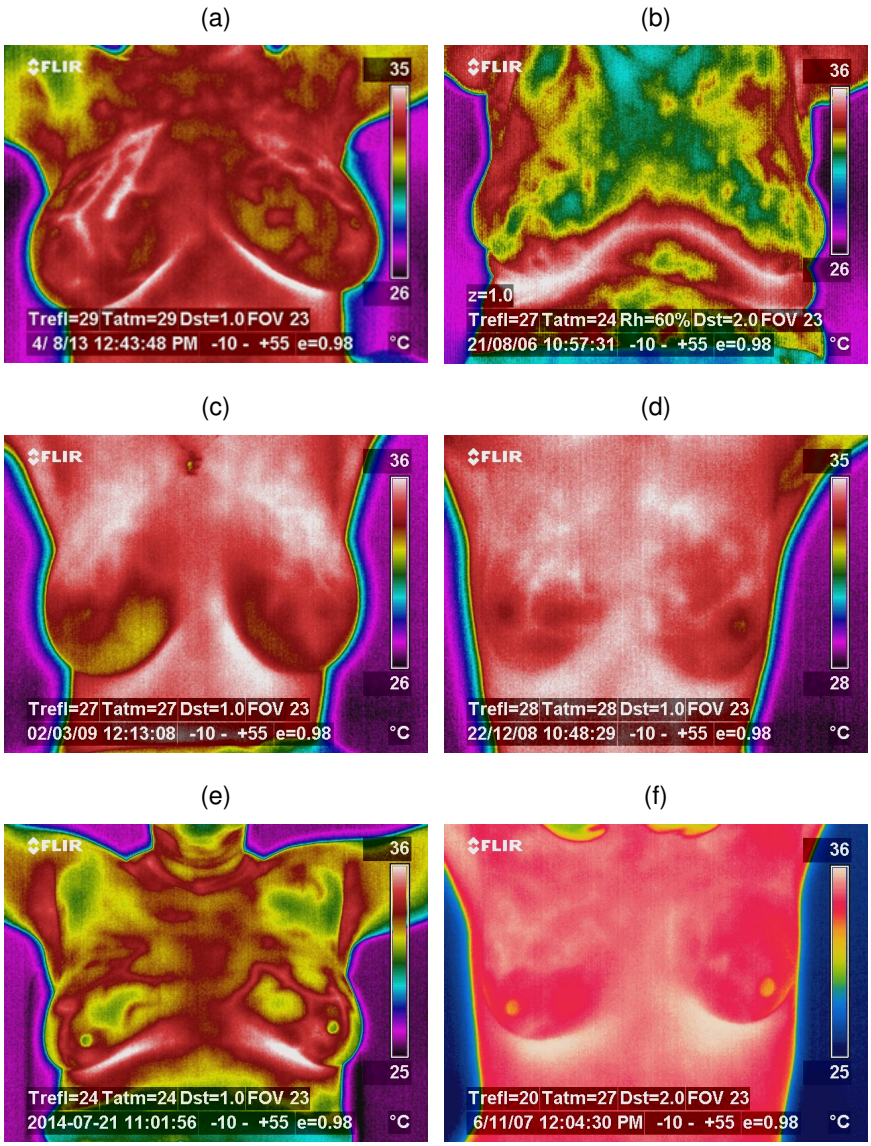


Fonte: O Autor (2022).

câmera e a paciente.

Durante a aquisição das imagens duas séries de imagens são obtidas. Uma série com distância fixa entre a câmera e a paciente e a outra com distância variável. A segunda série de imagens é feita para que possa ser analisada visualmente por médicos, de modo que eles possam observar se há alguma anormalidade na distribuição de temperatura, enquanto que a primeira série é feita para ser utilizada em métodos de processamento digital de imagens. Desta maneira, no protocolo de aquisição de imagens a distância fixa são realizadas as seguintes imagens: frontal de ambas as mamas (T1), frontal de ambas as mamas com as mãos para cima se apoiando no suporte superior do aparato (T2), lateral externa da mama direita e esquerda (LEMD e LEME) e lateral interna da mama direita e

Figura 4 – Imagens de termografia de mama de pacientes com lesão benigna.

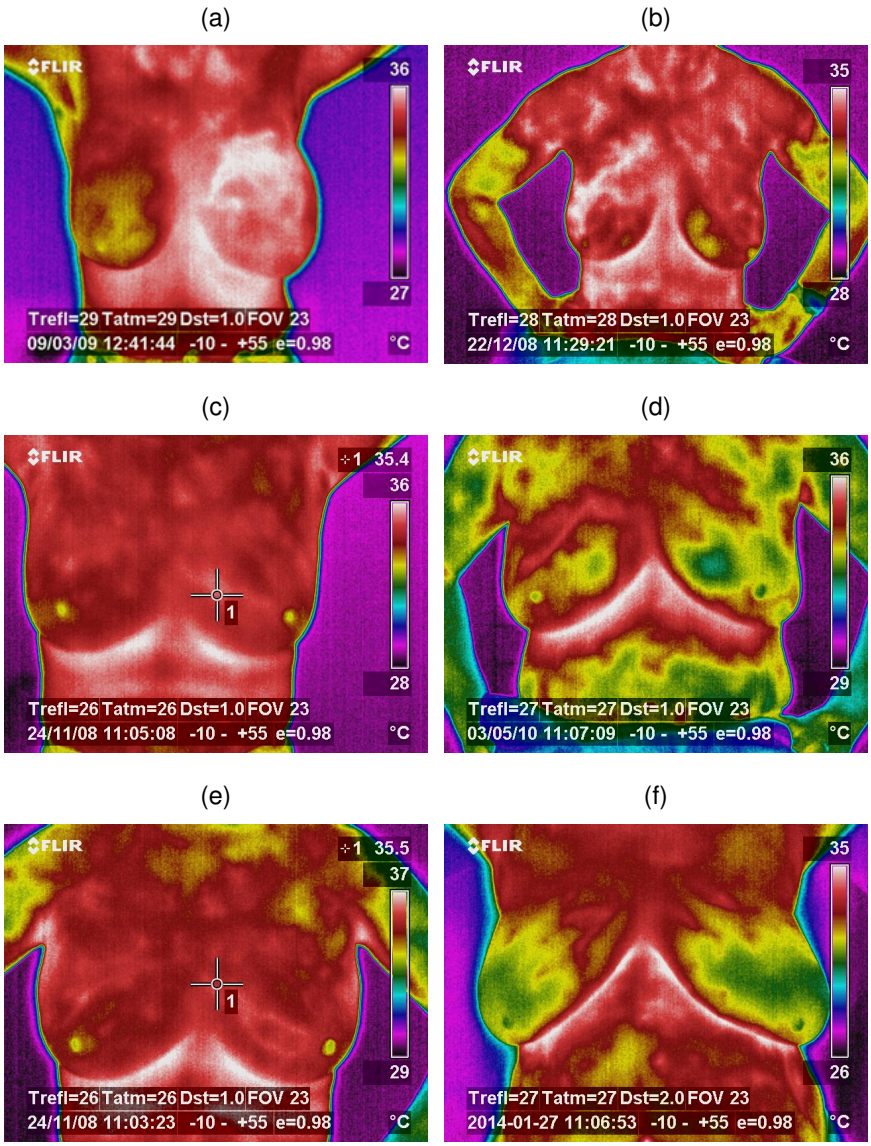


Fonte: O Autor (2022).

esquerda (LIMD e LIME). Na Figura 8 são mostrados exemplos de imagens adquiridas pelo grupo de pesquisa da UFPE.

As imagens adquiridas foram obtidas utilizando uma câmera termográfica da marca FLIR Systems, modelo ThermaCAMTM S45. Essa câmera possui um campo de visão de $24°\times18°/0{,}3$ m, resolução espacial de 1,3 mrad, detector do tipo matriz de plano focal (FPA), microbolômetro não resfriado, 320×240 *pixels*; amplitude espectral de 7,5-13 μm; amplitudes de temperaturas padrão disponíveis: -40 ℃ a 120 ℃, -10 ℃ e $+55$ ℃, 0 ℃ a 500 ℃, 350 ℃ a 1500 ℃ (OLIVEIRA, 2012). A escala utilizada para a termografia de mama é a -10 ℃ e $+55$ ℃. A sensibilidade térmica da câmera é de 0,06 ℃ e sua precisão de ±1 ℃ para o intervalo de temperatura utilizado (OLIVEIRA, 2012).

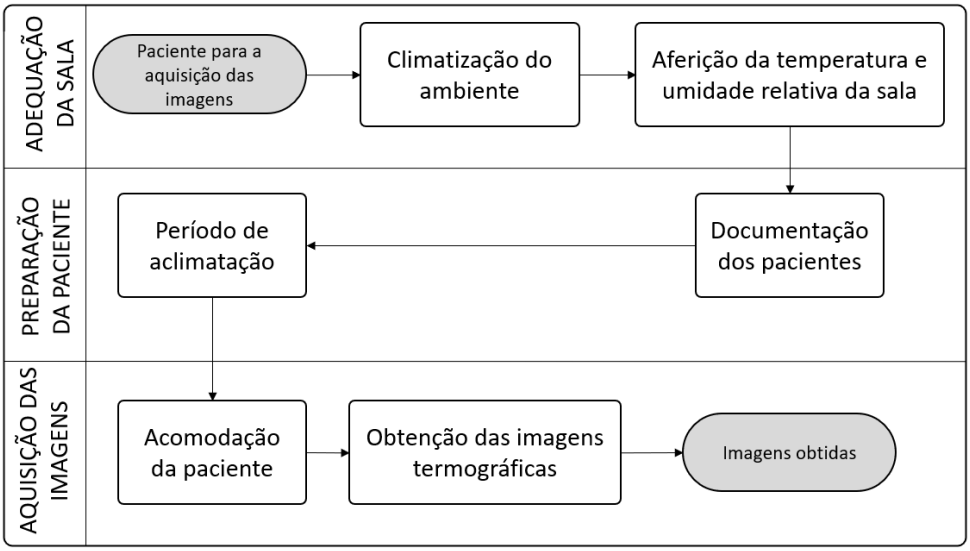
Figura 5 – Imagens de termografia de mama de pacientes com lesão maligna.



Fonte: O Autor (2022).

De acordo com a avaliação de médicos especialistas, as imagens foram separadas em quatro grupos: sem lesão, com cisto, com lesão benigna e com lesão maligna. Exemplos dessas imagens são dados nas Figuras 2, 3, 4 e 5. Os diagnósticos das pacientes foram realizados de acordo com exames específicos para cada situação. As imagens de pacientes sem lesão foram rotuladas desta forma ao considerar os exames de mamografia e ultrassonografia das pacientes, classificados como BI-RADS 1, sem achados (SILVA, 2015). Já para as imagens avaliadas com cistos, os diagnósticos das as pacientes foram realizados através de punção aspirativa com agulha fina (PAAF) ou ultrassonografia. Por fim, no caso de lesões benigna e maligna o diagnóstico se deu através de biópsias (SILVA, 2015). Para todos os casos, o padrão ouro para o diagnóstico foi o exame clínico associado a biópsia.

Figura 6 – Fluxograma do protocolo de aquisição de imagens térmicas de mama.



Fonte: O Autor (2022).

Figura 7 – Aparato para aquisição de imagens termográficas de mama.

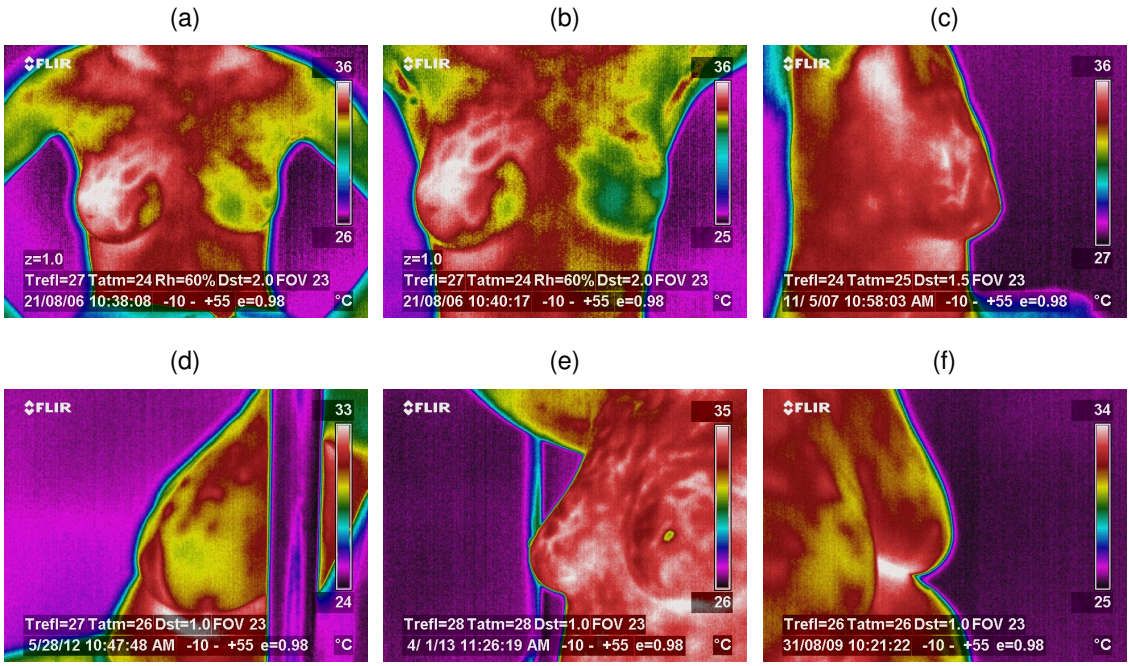
Fonte: Oliveira (2012)

Porém a realização de biópsias não ocorreu nos casos onde não havia a necessidade para o tal, como é o caso das pacientes diagnósticas como sem lesão através da mamografia ou ultrassonografia.

2.2 Redes neurais artificiais

Inspirados em redes de neurônios do cérebro, as Redes Neurais Artificiais (RNAs) são uma modalidade de algoritmos da Inteligência Artificial capazes de aprender. Entende-se como aprender a capacidade de mapear determinadas entradas em determinadas saídas, com um erro aceitável. Para este processo é necessário fornecer para a rede um conjunto de dados referente ao problema em que se deseja resolver utilizando a rede.

Figura 8 – Exemplo das posições de aquisição das imagens por paciente a uma distância fixa: (a) T1, (b) T2, (c) LEMD, (d) LEME, (e) LIMD e (f) LIME



Fonte: O Autor (2022).

Os primeiros tipos de redes neurais utilizam funções de mapeamento linear para separar dados distintos. Para essas redes, denomina-se de neurônio a unidade de processamento de informação que é fundamental para a operação de uma rede neural (HAYKIN, 2001). Um modelo de um neurônio k contendo N entradas é dado na Figura 9. Segundo Haykin (2001), neste modelo, destacam-se os seguintes parâmetros:

1. Os sinais de entradas dos neurônios representados pelos termos x_j para $j = 1, 2, \ldots, N$;

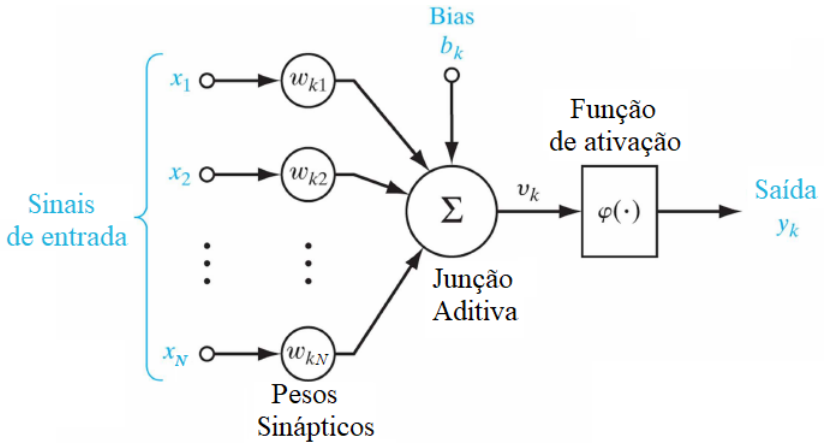
2. Os pesos sinápticos do neurônio k (dados por w_{kj}), onde cada entrada x_j será multiplicada pelo seu respectivo peso w_{kj};

3. A junção aditiva ou somador que irá atuar como um combinador linear ao somar os termos $w_{kj} \cdot x_j$;

4. O termo bias b_k que será somado ao resultado obtido pelo somador. O termo bias tem por função aumentar ou diminuir a entrada líquida da função de ativação;

5. A função de ativação $\varphi(\cdot)$, a qual será aplicada à soma do bias com o resultado do somador de modo a delimitar a saída (y_k) do neurônio k.

Figura 9 – Modelo de um neurônio.



Fonte: Adaptado de Haykin (2001).

Se denotarmos por v_k a junção do termo bias com a saída do somador, pode-se expressar matematicamente a relação entre a saída y_k do neurônio k em função das entradas x_j através das Equações 2.3 e 2.4.

$$v_k = \sum_{j=1}^{N} w_{kj} x_j + b_k \tag{2.3}$$

$$y_k = \varphi(v_k) \tag{2.4}$$

Um outro modelo de neurônio bastante utilizado é mostrado na Figura 10, onde o bias é reposicionado. Nesse modelo o bias passar a ser incorporado ao peso da entrada x_0 que é sempre mantida igual a 1.

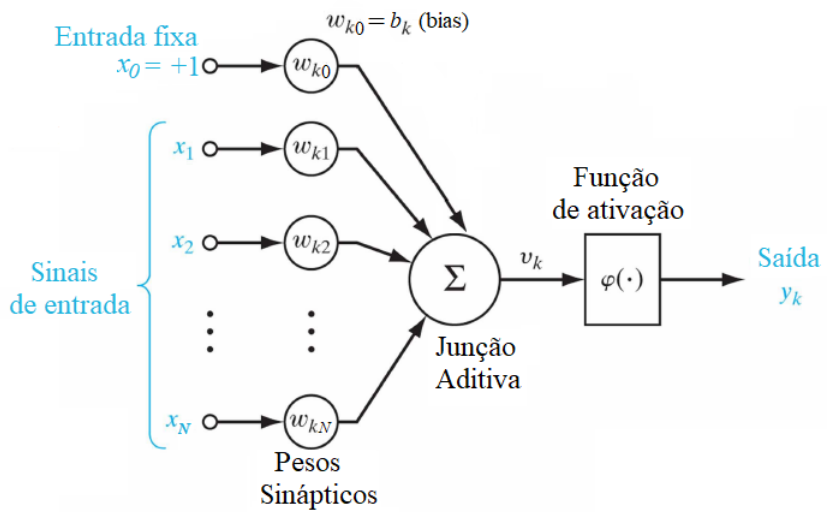
Apesar de serem diferentes, os modelos apresentados nas Figuras 9 e 10 são equivalentes matematicamente (HAYKIN, 2001). Para o segundo modelo apresentado a saída do neurônio é dada pelas Equações 2.5 e 2.6, onde $x_0 = +1$ e $w_{k0} = b_k$.

$$v_k = \sum_{j=0}^{N} w_{kj} x_j \tag{2.5}$$

$$y_k = \varphi(v_k) \tag{2.6}$$

A função de ativação pode assumir várias formas diferentes. Na Figura 11 são dados quatro formas comumente utilizadas na Inteligência Artificial. A função dada pela

Figura 10 – Modelo alternativo de neurônio.



Fonte: Adaptado de Haykin (2001).

Figura 11a é chamada de função limiar que assume valores 0 se sua entrada for negativa, ou 1, se a entrada for maior ou igual a zero. Sua representação matemática é dada na Equação 2.7. Na Figura 11b é mostrado o gráfico da função linear por partes que possui uma zona de transição linear entre os extremos 0 e 1 da função (Equação 2.8). A função sigmoide, dada em 11c, é caracterizada por ter seu gráfico em forma de s (HAYKIN, 2001). Na Equação 2.9 é dado um tipo de Função Sigmoide chamada de função logística, onde o parâmetro a é chamado de parâmetro de inclinação. Por fim, a função retificada linear (ou do inglês *Rectified Linear Activation Function* - ReLU) (NAIR; HINTON, 2010; KRIZHEVSKY; SUTSKEVER; HINTON, 2012), muito utilizada nas *Deep convolutional neural networks* (ver Secção 2.4), sua equação é dada em 2.10 e o seu gráfico é dado na Figura 11d.

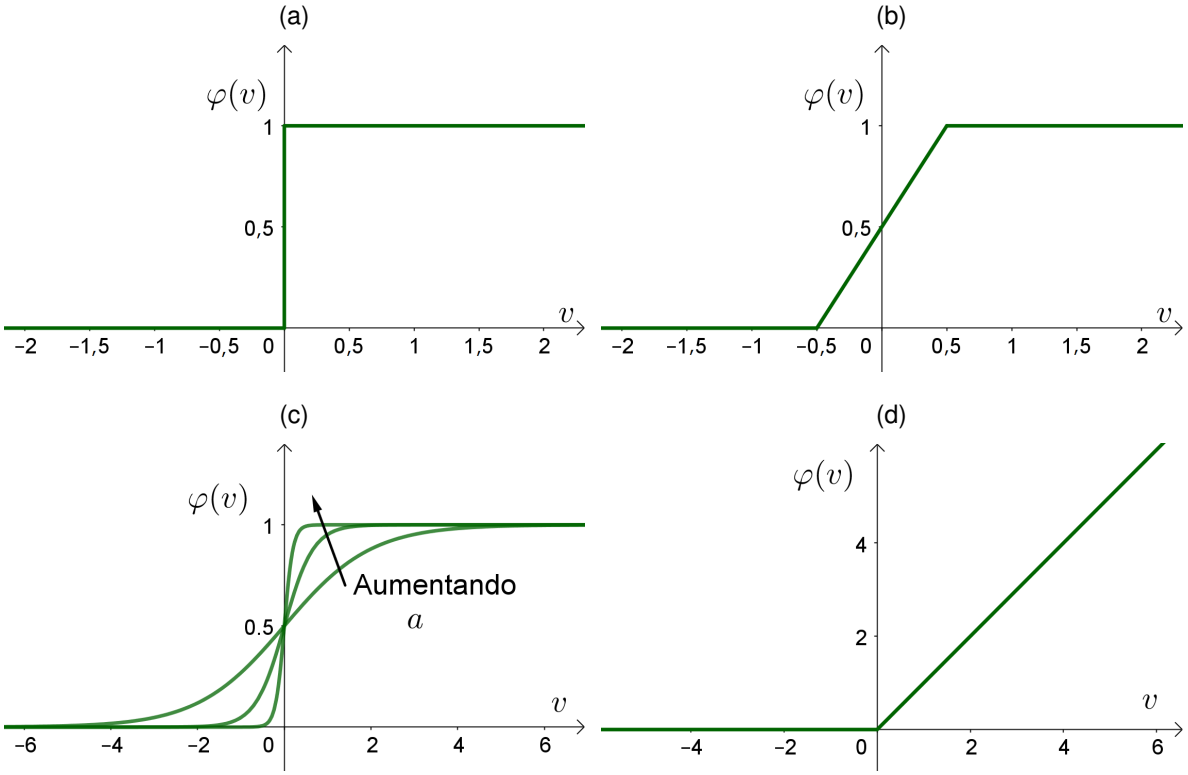
$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \tag{2.7}$$

$$\varphi(v) = \begin{cases} 1, & v \geq +\frac{1}{2} \\ v, & +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & v \leq -\frac{1}{2} \end{cases} \tag{2.8}$$

$$\varphi(v) = \frac{1}{1 + exp(-av)} \tag{2.9}$$

$$\varphi(v) = max(0, v) \tag{2.10}$$

Figura 11 – Exemplos de funções de ativação. (a) Função Limiar (ou de Heaviside), (b) função Linear por Partes e (c) Função Logística (Sigmoide), (d) ReLU.



Fonte: O Autor (2022).

As funções de ativação apresentadas anteriormente possuem como imagem valores entre 0 e 1, mas há também uma função de ativação comumente utilizada a qual assume valores entre -1 e 1. Esta função é uma modificação da função limiar, a qual é denominada função sinal (HAYKIN, 2001), definida por:

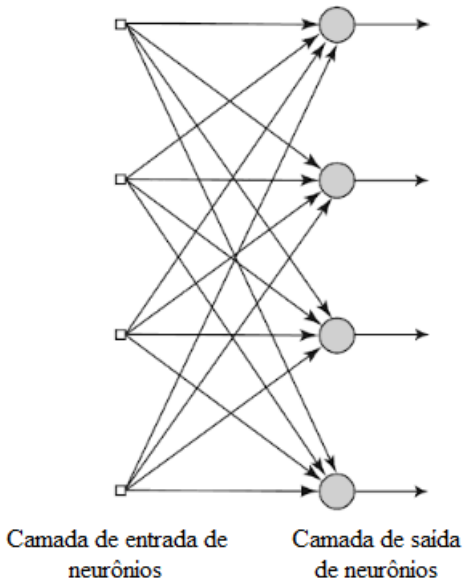
$$\varphi(v) = \begin{cases} 1, & v > 0 \\ 0, & v = 0 \\ -1, & v < 0 \end{cases} \tag{2.11}$$

Como citado anteriormente, o neurônio é a unidade de uma RNA, a seguir apresenta-se como alguns neurônios podem ser agrupados de modo a formar uma rede. O modelo mais simples de RNA é a rede de neurônios de camada única como aquela mostrada na Figura 12. Nessa arquitetura, tem-se uma camada de entrada conectada diretamente à camada de saída formada por neurônios. Na figura é mostrada uma camada de entrada

possuindo quatro entradas as quais estão conectadas a quatro neurônios que formam a camada de saída da rede. A camada de entrada não é contada pois nela não é feita nenhuma operação (BRAGA; CARVALHO; LUDERMIR, 1998; HAYKIN, 2001). Uma rede de camada única pode ser treinada para problemas de classificação de duas classes diferentes, por exemplo.

O propósito do treinamento é fazer com que a rede extraia informações relevantes de padrões dos dados apresentados para a mesma, de modo que ela crie uma representação própria para o problema (BRAGA; CARVALHO; LUDERMIR, 1998). Como por exemplo, nas funções de ativação anteriores (que possuem como imagem valores no intervalo [0, 1]), pode-se interpretar a saída 0 como uma dada classe e a saída 1 para uma outra classe. A este tipo de problema se chama de problema de classificação binária. Neste trabalho, especificamente, um exemplo de classificação binária é concluir se existe alguma lesão, ou não, em uma imagem de termografia de mama.

Figura 12 – Rede de neurônios de camada única.

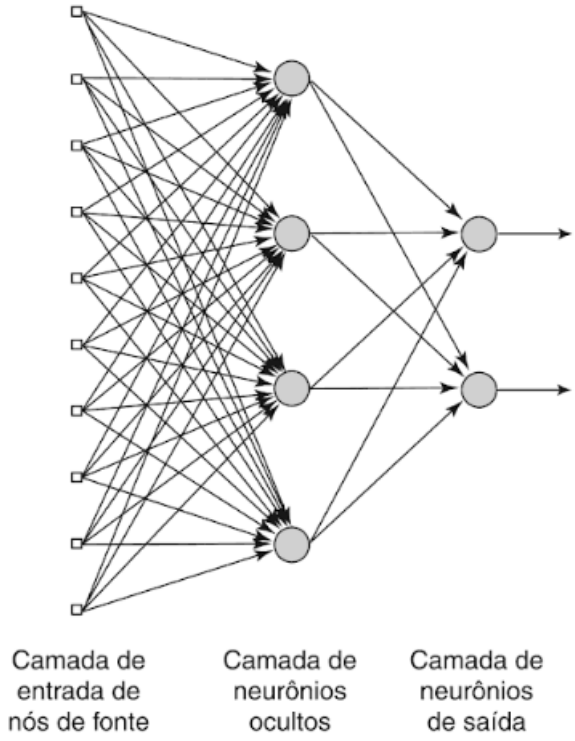


Fonte: Adaptado de Haykin (2001).

Um outro tipo de arquitetura para RNAs é o modelo de múltiplas camadas como é mostrado na Figura 13. Essa arquitetura é caracterizada por possuir várias camadas de neurônios conectadas entre si. Assim, a camada de entrada estará conectada a uma camada de neurônios denominada de camada intermediária ou camada escondida, esta então estará conectada a outra camada escondida e assim por diante. Por fim, na última camada, se tem a camada de saída. Em particular, na Figura 13 é mostrado uma rede de duas camadas, onde se tem uma camada intermediária e uma camada de saída. Observa-

se na Figura 13 que cada nó (uma outra denominação para neurônio) de uma camada da rede está conectado a todos os nós da camada seguinte, nesse caso se diz que a rede é totalmente conectada. Se alguns elos estiverem faltando na rede, diz-se que a rede é parcialmente conectada (HAYKIN, 2001).

Figura 13 – Rede de neurônios de camada dupla.



Fonte: Adaptado de Haykin (2001).

O treinamento de uma RNA tem como objetivo ajustar os pesos dos neurônios para que sua saída esteja de acordo com a entrada inserida. Esse tipo de abordagem é chamado de treinamento supervisionado. Nessa abordagem parte do conjunto de dados do problema será fornecida à rede para treinamento. Além disso, é fornecida também a saída desejada para cada entrada dada. Essa parcela do conjunto de dados usada para treinamento é chamada de conjunto de treinamento, enquanto que a outra parcela é chamada de conjunto de teste ou conjunto de validação, pois após o treinamento, o conjunto de teste é utilizado para avaliar o desempenho da rede. Nesse momento a saída desejada não é mais fornecida. Neste trabalho, onde serão usadas redes neurais para problemas de classificação, esse será o tipo de aprendizagem utilizado.

Uma outra abordagem de treinamento utilizada é o treinamento não-supervisionado. Nessa modalidade a saída esperada não é fornecida para a rede. O objetivo é que a rede encontre padrões entre os dados de entradas para diferenciá-los entre si e criar novas

classes.

Algumas nomenclaturas que são largamente utilizadas na Inteligência Artificial são a de instância e atributos. Para um conjunto de dados, chama-se de instância um exemplo ou ponto que compõe o conjunto, e dado um ponto, denominam-se atributos as componentes do ponto. Por exemplo, para um conjunto de dados que contém 100 pontos do $\mathbb{R}^3$ tem-se 100 instâncias, cada uma com 3 atributos.

As redes apresentadas nas Figuras 12 e 13 são classificadas como redes alimentadas adiante (ou do inglês *feedforward neural networks*), pois suas conexões entre neurônios são feitas sempre da camada de entrada em direção a camada de saída, mas nunca o contrário (HAYKIN, 2001). Neste trabalho foram utilizadas apenas redes alimentadas adiante.

Um conceito importante sobre classificadores é o de generalização, que é a capacidade do classificador prever corretamente a classe de novos dados. Vale a pena destacar os conceitos de superajustamento (*overfitting*) e subajustamento (*underfitting*). Diz-se que um classificador está superajustado quando este se especializa nos dados de treinamento, obtendo um bom desempenho, mas que se sai malsucedido ao ser confrontado com novos dados. Enquanto que o subajustamento refere-se quando o classificador obtém baixa taxa de acertos no conjunto de treinamento, que pode acontecer pelo fato do conjunto de treinamento ser pouco representativo, ou pelo fato da rede não está dimensionada de forma adequada para o problema (LORENA; CARVALHO, 2007).

Segundo Braga, Carvalho e Ludermir (1998), para se obter uma boa generalização com RNAs, deve-se fornecer para a rede a maior quantidade possível de informação a respeito do problema a ser solucionado, isto é, fornecer o maior conjunto de dados possível. Para isso, também é necessário o uso de uma grande quantidade de nós na rede. Em contrapartida, por razões de complexidade computacional, deve-se buscar reduzir ao mínimo o número de nós e a quantidade de conexões entre eles (BRAGA; CARVALHO; LUDERMIR, 1998).

2.2.1 Perceptron de Múltiplas Camadas

O psicólogo ROSENBLATT foi um dos pioneiros da concepção das redes neurais artificiais ao propor, em 1958, o modelo de perceptron para a aprendizagem supervisionada.

O perceptron é a forma mais simples de rede neural usada para classificações binárias de padrões linearmente separáveis. Ele consiste de um único neurônio com pesos sinápticos ajustáveis e bias como mostrado na Figura 9. Também é possível acrescentar mais neurônios na camada de saída do perceptron para torná-lo capaz de classificar problemas com mais de duas classes (HAYKIN, 2001).

Como uma generalização do perceptron de camada única, o perceptron de múltiplas camadas (MLP, *Multilayer Perceptron*) é uma rede de múltiplas camadas completamente conectadas como aquela mostrada na Figura 13. O acréscimo de camadas intermediárias fornece à rede a capacidade de classificar problemas difíceis e de maior grau de complexidade, diferentemente dos problemas linearmente separáveis, os únicos que o perceptron de camada única consegue resolver. Teoricamente, com uma camada intermediária é possível aproximar a solução da rede a qualquer função contínua, enquanto que, duas camadas intermediárias são suficientes para aproximar qualquer função matemática (CYBENKO, 1988; CYBENKO, 1989).

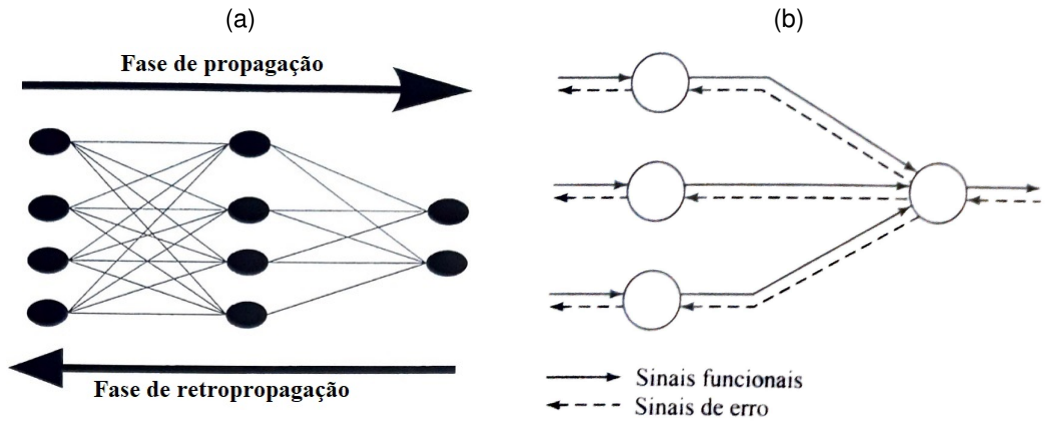
A MLP pode ser usada tanto para problemas de classificação quanto de regressão linear. O principal algoritmo para o treinamento de redes MLPs é o algoritmo de retropropagação de erro (*error backpropagation*). Baseado em gradiente descendente, o método visa minimizar iterativamente o erro entre a saída da rede obtida e a saída desejada (HAYKIN, 2001; AFFONSO et al., 2010). Sendo um método supervisionado, sua aplicação se dá através da retro-propagação dos valores da saída para a entrada, e então os pesos são ajustados e uma nova saída é obtida e o procedimento é repetido (AFFONSO et al., 2010).

Para utilizar esse algoritmo é necessário que a função de ativação seja contínua, diferenciável e, de preferência, não decrescente. Além disso, como é provado em Braga, Carvalho e Ludermir (1998), a função de ativação utilizada em uma MLP precisa ser não linear, pois uma rede multicamadas que utiliza uma função de ativação linear é equivalente a uma rede de única camada. A função de ativação mais comumente utilizada para as MLPs é a função sigmoide logística (BRAGA; CARVALHO; LUDERMIR, 1998; HAYKIN, 2001).

O treinamento de uma rede MLP através da retropropagação de erro que acontece em duas fases: propagação e retropropagação. Na fase propagação se obtém uma saída para um dado padrão de entrada. Na retroprogação é utilizada a saída desejada e a saída obtida na fase de propagação para atualizar os pesos de suas conexões (BRAGA;

CARVALHO; LUDERMIR, 1998; HAYKIN, 2001). Na Figura 14a é mostrado os sentidos de fluxo dos dados nas duas fases.

Figura 14 – (a) Esquemático do fluxo do algoritmo de treinamento de retropropagação de erro. (b) Direção do fluxo dos sinais funcionais e de erro.



Fontes: (a) Adaptado de Braga, Carvalho e Ludermir (1998); (b) Haykin (2001).

Segundo Haykin (2001), na fase de propagação são os denominados sinais funcionais que são transmitidos pela rede. Os sinais funcionais recebem esse nome por serem de origem da aplicação dos pesos e das funções de aplicação a um sinal de entrada da rede. Já na fase de retropropagação são os sinais de erro que fazem o percurso inverso. Na Figura 14b é mostrado os fluxos dos sinais funcionais e de erro. As etapas das fases de propagação e retropropagação estão explicitadas nos Algoritmos 1 e 2. No Algoritmo 3 é explicitado o método da retropropagação de error em função de suas respectivas fases. Como critério de parada para o algoritmo é comum se utilizar um número máximo de iterações (também chamadas de épocas), ou após o erro da rede ficar menor do que um valor estabelecido.

Algoritmo 1 Fase de propagação

1: Inserir uma entrada na camada de entrada da rede, C^0;
2: **para** cada camada C^i **fazer**
3: Calcular as saídas dos neurônios da camada C^i ($i > 0$) e aplique como entrada nos neurônios da camada C^{i+1};
4: **fim para**
5: Comparar as saídas produzidas com as saídas desejadas.

Fonte: Braga, Carvalho e Ludermir (1998).

Para a atualização dos pesos sinápticos o algoritmo utiliza como função de custo a ser minimizada a soma dos erros quadráticos para os neurônios da camada de saída,

Algoritmo 2 Fase de retropropagação

1: Calcular o erro da camada de saída da rede
2: **para** cada camada C^i, começando da última até a primeira **fazer**
3: Ajustar os pesos dos neurônios da camada C^i de modo a reduzir seus erros;
4: Calcular o erro dos neurônios da camada C^{i-1} a partir do erros da camada C^i, ponderado pelos pesos das conexões entre eles;
5: **fim para**

Fonte: Braga, Carvalho e Ludermir (1998).

Algoritmo 3 Retroprogação de erro

1: Inicializar pesos e parâmetros;
2: **repetir**
3: **para** cada par entrada/saída desejada do conjunto de treinamento **fazer**
4: Executar a fase de propagação;
5: Comparar a saída obtida com a saída desejada;
6: Executar a fase de retroprogação para atualizar os pesos dos nós;
7: **fim para**
8: **até** que algum critério de parada seja atingido.

Fonte: Braga, Carvalho e Ludermir (1998).

também chamada de energia (E), dada pela Equação 2.12.

$$E = \frac{1}{2} \sum_{p} \sum_{i=1}^{N_S} (d_i^p - y_i^p)^2 \tag{2.12}$$

onde p é o número de classes do problema, N_S é o número de neurônios na camada de saída, d_i é a i-ésima saída desejada com respeito à uma entrada x_i e y_i é a i-ésima saída obtida pela rede. A saída de um neurônio j qualquer pode ser obtida pela expressão:

$$y_j^p = \varphi_j(v_j^p) \tag{2.13}$$

onde $v_j^p = (\sum_{i=1}^{n} x_i^p \cdot w_{ji})$ e w_{ji} é o peso da conexão entre a entrada x_i^p e o neurônio j. A atualização dos valores dos pesos para diminuir o erro, E, é baseada na regra delta onde a variação dos pesos é diretamente proporcional com o gradiente descendente do erro com relação ao peso (BRAGA; CARVALHO; LUDERMIR, 1998), assim, pode-se expressar que:

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} \tag{2.14}$$

onde Δw_{ji} representa o quanto o peso w_{ji} deve ser variado para minimizar o erro. Após

algumas manipulações matemática, e aplicação de regras do cálculo diferencial[3], encontra-se que a variação do erro é dado por:

$$\Delta w_{ji} = \eta \delta_j x_i \tag{2.15}$$

onde η é a taxa de aprendizagem (um parâmetro do método), δ_j representa o erro do neurônio j, o seu valor é dado pela Equação 2.16 se o neurônio está na camada de saída da rede, ou é dado pela Equação 2.17 se o neurônio está numa camada intermediária. Nesta expressão o índice l é usado para representar um neurônio na camada de saída.

$$\delta_l = (d_l - y_l) \cdot \varphi'(v_l) \tag{2.16}$$

$$\delta_j = \varphi'(v_j) \cdot \sum_l \delta_l w_{lj} \tag{2.17}$$

Assim, analisando a Equação 2.16 se percebe que os neurônios da camada de saída possuem a real medida do erro cometido por eles, pois no termo δ_l compara-se justamente a saída desejada com a saída do neurônio em questão. Enquanto que, analisando a Equação 2.17 se percebe que um neurônio de uma camada intermediária recebe apenas uma estimativa do erro dos neurônios da camada posterior a dele, pois o seu erro vai depender do erro dos neurônios posteriores. E assim, denotando por t a iteração do processo de minimização do erro, tem-se que o valor do peso será atualizado pela seguinte expressão:

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_j(t) x_i(t) \tag{2.18}$$

Como dito anteriormente, o uso de mais de duas camadas intermediárias pode ser recomendado para facilitar o treinamento da rede. De contrapartida, não se deve utilizar um número grande de camadas intermediárias pois durante o processo de treinamento, o erro medido será propagado de camada em camada se tornando cada vez mais impreciso, isto é, a última camada intermediária recebe uma estimativa do erro da camada de saída e a penúltima camada intermediária recebe a estimativa da estimativa deste erro, e assim por

[3] Para mais detalhes sobre como é feita a atualização dos pesos da rede MLP através do gradiente sugere-se a leitura dos textos Johansson, Dowla e Goodman (1991), Riedmiller (1994), Braga, Carvalho e Ludermir (1998) e Haykin (2001).

diante (BRAGA; CARVALHO; LUDERMIR, 1998).

Segundo Huang, Zhu e Siew (2004), o algoritmo de retropropagação de erro possui algumas desvantagens, citadas a seguir.

1. Quando a taxa de aprendizagem η é baixa demais, o processo de aprendizagem acontece de forma demasiada lenta, contudo se η for muito alta o algoritmo se torna instável;
2. A retropropagação de erro é suscetível a cair em mínimos locais;
3. A MLP pode ser superajustada pelo algoritmo resultando em baixa generalização, sendo necessário o uso de validações e critérios de parada adequados para a minimização da função de custo;
4. Na maioria das aplicações, o processo de aprendizagem baseado em gradiente descendente é lento.

Desta maneira, para aumentar a velocidade de aprendizagem e diminuir a chance de cair em um mínimo local, inclui-se o termo *momentum* dado por: $\alpha(w_{ji}(t) - w_{ji}(t-1))$ (BRAGA; CARVALHO; LUDERMIR, 1998; HAYKIN, 2001). Assim, a equação para a atualização dos valores do pesos após a inserção do *momentum* é:

$$w_{ji}(t+1) = w_{ji}(t) + \eta\delta_j(t)x_i(t) + \alpha(w_{ji}(t) - w_{ji}(t-1)) \tag{2.19}$$

Durante o processo de atualização dos pesos, as entradas são inseridas de modo aleatório. Existem duas formas de atualizar os pesos de acordo com a frequência de atualização. São os modos sequencial e por lote. No modo sequencial os pesos da rede são atualizados a cada nova entrada inserida. Já no modo por lote, os pesos são atualizados apenas quando todas as entradas estiverem sido inseridas na rede. O modo sequencial tem algumas vantagens práticas com relação ao modo por lote por necessitar menor uso de memória, por deixar a aprendizagem mais aleatória e por ser mais rápida. Por outro lado, o modo por lote é mais estável (BRAGA; CARVALHO; LUDERMIR, 1998; HAYKIN, 2001).

2.2.2 Máquina de Aprendizado Extremo

A Máquina de Aprendizado Extremo (ELM, *Extreme Learning Machine*) é uma abordagem de treinamento para redes neurais *feedforward* com no mínimo uma camada escondida, caracterizada por gerar os pesos de entrada aleatoriamente, e determinar os pesos da camada de saída de forma analítica através de um sistema de equações lineares. Diferente dos métodos em que todos os parâmetros da rede como pesos e bias precisam ser ajustados como a retroprogação de erro das MLPs. Por ser um método analítico, não corre o risco de ficar preso em um mínimo local. Teoricamente, as ELMs possuem boa generalização associada a uma aprendizagem rápida, características estas que podem ser relevantes em aplicações que demandam um treinamento intenso (HUANG; ZHU; SIEW, 2004; HUANG; ZHU; SIEW, 2006).

Nas próximas duas seções serão abordados alguns conceitos importantes e necessários para o entendimento das ELMs. Em seguida, o seu processo de treinamento é abordado.

A. Inversa generalizada de Moore-Penrose

Definição: Uma matriz $\boldsymbol{G}$ de ordem $N \times M$ é uma inversa generalizada de Moore-Penrose de uma matriz $\boldsymbol{A}$ de ordem $M \times N$, se forem satisfeitas as condições:

1. $\boldsymbol{AGA} = \boldsymbol{A}$
2. $\boldsymbol{GAG} = \boldsymbol{G}$
3. $(\boldsymbol{AG})^T = \boldsymbol{AG}$
4. $(\boldsymbol{GA})^T = \boldsymbol{GA}$

Daqui em diante, a inversa generalizada de Moore-Penrose de uma matriz $\boldsymbol{A}$ será representada por $\boldsymbol{A}^+$. A matriz $\boldsymbol{A}^+$ pode ser definida mesmo se $\boldsymbol{A}$ for singular[4] ou até mesmo se não for quadrada (HUANG; ZHU; SIEW, 2004).

[4] Uma matriz quadrada é dita singular quando não admite uma inversa.

B. Solução de sistemas lineares por mínimos quadrados de norma mínima

Seja $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$ um sistema linear, diz-se que $\hat{\boldsymbol{x}}$ é uma solução de mínimos quadrados, se:

$$||\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{y}|| = \min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}|| \tag{2.20}$$

onde $||\cdot||$ representa a norma euclidiana.

Definição: um vetor $\boldsymbol{x}_0 \in \mathbb{R}^N$ é dito ser uma solução de mínimos quadrados de norma mínima de um sistema linear $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$ se para qualquer $\boldsymbol{y} \in \mathbb{R}^M$

$$||\boldsymbol{x}_0|| \leq ||\boldsymbol{x}||, \ \forall x \in \{\boldsymbol{x} : ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}|| \leq ||\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}||, \ \forall \boldsymbol{z} \in \mathbb{R}^N\} \tag{2.21}$$

Significa que $\boldsymbol{x}_0$ é uma solução de mínimos quadrados de norma mínima se for a solução de mínimos quadrados de menor norma do sistema (HUANG; ZHU; SIEW, 2004). Uma vez tendo definido tal termo, pode-se enunciar um importante teorema para a teoria das máquinas de aprendizado extremo.

Teorema: Seja uma matriz $\boldsymbol{G}$ tal que $\boldsymbol{G}\boldsymbol{y}$ é a solução de mínimos quadrados de norma mínima do sistema $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$. Então é suficiente e necessário que $\boldsymbol{G} = \boldsymbol{A}^+$.

C. Treinamento de uma Máquina de Aprendizado Extremo

Considere uma rede neural com uma camada escondida contendo $\widetilde{N}$ neurônios escondidos submetida a um conjunto de treinamento $\{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^N$, onde $\boldsymbol{x}_i \in \mathbb{R}^N$ e $\boldsymbol{t}_i \in \mathbb{R}^M$. Tal rede pode ser modelada pela equação:

$$\sum_{i=1}^{\widetilde{N}} \beta_i \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x}_j + b_i) = \boldsymbol{y}_j, \ j = 1, \ \ldots, N \tag{2.22}$$

onde φ é a função de ativação, $\boldsymbol{w}_i$ é o vetor de pesos que conecta o i-ésimo neurônio escondido aos neurônios de entrada, β_i é o vetor de pesos que conecta o i-ésimo neurônio aos neurônios de saída e b_i é o bias do i-ésimo neurônio escondido.

Se o número de neurônios da camada escondida é igual ao de pontos do conjunto de treinamento, $\widetilde{N} = N$, tal rede neural com uma camada escondida pode aproximar as N amostras sem nenhum erro, ou seja, $\sum_{i=1}^{\widetilde{N}} ||\boldsymbol{y}_j - \boldsymbol{t}_j|| = 0$ (HUANG; ZHU; SIEW, 2004).

Sendo assim, existe β_i, $\boldsymbol{w}_i$ e b_i tais que:

$$\sum_{i=1}^{\widetilde{N}} \beta_i \varphi(\boldsymbol{w}_i \cdot \boldsymbol{x}_j + b_i) = \boldsymbol{t}_j,\ j = 1,\ \ldots, N \tag{2.23}$$

A Equação 2.23 representa um sistema de N equações que podem ser reescritas da seguinte forma:

$$\boldsymbol{H}\beta = \boldsymbol{T} \tag{2.24}$$

onde

$$\begin{aligned} &\boldsymbol{H}(\boldsymbol{w}_1,\ \ldots,\ \boldsymbol{w}_{\widetilde{N}},\ b_1,\ \ldots,\ b_{\widetilde{N}},\ \boldsymbol{x}_1,\ \ldots,\ \boldsymbol{x}_{\widetilde{N}}) \\ &\qquad = \begin{bmatrix} \varphi(\boldsymbol{w}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & \varphi(\boldsymbol{w}_{\widetilde{N}} \cdot \boldsymbol{x}_1 + b_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ \varphi(\boldsymbol{w}_1 \cdot \boldsymbol{x}_N + b_1) & \cdots & \varphi(\boldsymbol{w}_{\widetilde{N}} \cdot \boldsymbol{x}_{\widetilde{N}} + b_{\widetilde{N}}) \end{bmatrix}_{N \times \widetilde{N}} \end{aligned} \tag{2.25}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\widetilde{N}}^T \end{bmatrix}_{\widetilde{N} \times m} \quad \text{e} \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_{\widetilde{N}}^T \end{bmatrix}_{N \times m} \tag{2.26}$$

A matriz $\boldsymbol{H}$ é chamada de matriz de saída da camada escondida. A i-ésima coluna de $\boldsymbol{H}$ representa a saída do i-ésimo vetor com respeito as entradas $\boldsymbol{x}_j$, para $j = 1,\ \ldots, N$ (HUANG; ZHU; SIEW, 2004).

Na maioria das situações práticas o número de neurônios da camada escondida é muito menor do que o número de amostras, $\widetilde{N} \ll N$, de modo que $\boldsymbol{H}$ não é uma matriz quadrada e que os parâmetros β_i, $\boldsymbol{w}_i$ e b_i não existem (HUANG; ZHU; SIEW, 2004). Desta forma, para treinar uma rede neural de camada única é necessário encontrar a solução de mínimos quadrados $\hat{\beta}$ do sistema da Equação 2.24:

$$\begin{aligned} &||\boldsymbol{H}(\boldsymbol{w}_1,\ \ldots,\ \boldsymbol{w}_{\widetilde{N}},\ \boldsymbol{b}_1,\ \ldots,\ \boldsymbol{b}_{\widetilde{N}})\hat{\beta} - \boldsymbol{T}|| \\ &\qquad = \min_{\beta} ||\boldsymbol{H}(\boldsymbol{w}_1,\ \ldots,\ \boldsymbol{w}_{\widetilde{N}},\ \boldsymbol{b}_1,\ \ldots,\ \boldsymbol{b}_{\widetilde{N}})\beta - \boldsymbol{T}|| \end{aligned} \tag{2.27}$$

onde os pesos de entrada $\boldsymbol{w}_i$ e o biais da camada intermediária b_i são obtidos de maneira aleatória e são mantidos fixos durante todo o processo de aprendizagem. O objetivo da

Equação 2.27 é encontrar o valor $\hat{\beta}$ que minimiza o erro entre a saída da rede $\boldsymbol{H}\beta$ e a saída desejada $\boldsymbol{T}$. Além disso, a Equação 2.27 é uma reformulação da Equação 2.20 dedicada para esse problema. Segundo o teorema postulado na seção anterior, a solução de mínimos quadrados de norma mínima da Equação 2.27 é dada por:

$$\hat{\beta} = \boldsymbol{H}^{+}\boldsymbol{T} \tag{2.28}$$

Uma importante propriedade das máquinas de aprendizado extremo é que o menor erro de treinamento é obtido, não sendo susceptível a cair em um mínimo local feito as MLPs, por exemplo (HUANG; ZHU; SIEW, 2004). Além disso, a aprendizagem das ELMs garante um vetor de pesos de menor norma, pois:

$$||\hat{\beta}|| = ||\boldsymbol{H}^{+}\boldsymbol{T}|| \leq ||\beta||,\ \forall \beta \in \left\{\beta : ||\boldsymbol{H}\beta - \boldsymbol{T}|| \leq ||\boldsymbol{H}\boldsymbol{z} - \boldsymbol{T}||,\ \forall \boldsymbol{z} \in \mathbb{R}^{\widetilde{N}\times N}\right\} \tag{2.29}$$

Isto implica que a ELM possui melhor desempenho em generalização, pois segundo Bartlett (1997) uma rede neural treinada com baixos valores de pesos e com baixo erro quadrático possui boa capacidade de generalização. Essa é uma importante vantagem das ELMs, pois métodos de aprendizagem como a retroprogação de erro visa apenas minimizar o erro de treinamento sem considerar a magnitude dos pesos (HUANG; ZHU; SIEW, 2004). Por fim, a solução do processo de aprendizagem, dada pela solução de mínimos quadrados de norma mínima $\hat{\beta}$, é única (HUANG; ZHU; SIEW, 2004). No Algoritmo 4 é dado um resumo do processo de aprendizagem de uma máquina de aprendizagem extremo.

Algoritmo 4 Algoritmo da máquina de aprendizado extremo

Entrada: Um conjunto de treinamento $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{N}$, uma função de ativação φ, e $\widetilde{N}$ neurônios na camada escondida.

1: Atribua de forma aleatória os pesos de entrada $\boldsymbol{w}_i$ e bias b_i, $i = 1, \ldots, \widetilde{N}$;
2: Calcule a matriz dos pesos da camada intermediária $\boldsymbol{H}$;
3: Calcule a matriz dos pesos de saída β:

$$\beta = \boldsymbol{H}^{+}\boldsymbol{T} \tag{2.30}$$

onde $\boldsymbol{H}$, b e $\boldsymbol{T}$ são dados nas Equações 2.25 e 2.26.

Fonte: Huang, Zhu e Siew (2004).

As máquinas de aprendizado extremo podem ser estendidas para o uso de funções *kernels* (HUANG; SIEW, 2005; MICHE et al., 2009; CAMBRIA et al., 2013). Neste caso, a

modelagem da rede dada pela Equação 2.22 pode ser modificada para a seguinte maneira:

$$\sum_{i=1}^{\widetilde{N}} \beta_i K(\boldsymbol{x}, \boldsymbol{w}_i) = \boldsymbol{y}_j,\ j = 1,\ \ldots, N \tag{2.31}$$

onde $K : \mathbb{R} \to \mathbb{R}$ é a função *kernel* dos neurônios da camada escondida. Algumas das funções mais usadas são o *kernel* linear e o *kernel* gaussiano ou a função de base radial (RBF) dados na Equações 2.32 e 2.33 (HUANG; SIEW, 2005; MICHE et al., 2009; AZEVEDO et al., 2015):

$$K(\boldsymbol{x}, \boldsymbol{w}_i) = b_i + \boldsymbol{x} \cdot \boldsymbol{w}_i \tag{2.32}$$

$$K(\boldsymbol{x}, \boldsymbol{w}_i) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{w}_i||^2}{2\sigma_i^2}\right) \tag{2.33}$$

onde $\boldsymbol{w}_i$ e b_i são o vetor de pesos, e o bias do i-ésimo neurônio escondido, respectivamente e σ_i controla o raio da função gaussiana, para $i = 1, 2, \ldots, N$. Assim, a matriz de saída da camada escondida é dada então por:

$$\boldsymbol{H} = \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{w}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{w}_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_N, \boldsymbol{w}_1) & \cdots & K(\boldsymbol{x}_N, \boldsymbol{w}_{\widetilde{N}}) \end{bmatrix}_{N \times \widetilde{N}} \tag{2.34}$$

Nesta abordagem, o processo de aprendizado é realizado também através da matriz inversa de Moore-Penrose e o restante do processo de aprendizado é igual ao discutido para uma função de aptidão qualquer, exceto no Passo 1 do Algoritmo 4, onde além dos pesos $\boldsymbol{w}_i$ e bias b_i podem ser definidos outros parâmetros de forma aleatória como σ_i para um *kernel* RBF (HUANG; SIEW, 2005).

2.3 Máquina de Vetor de Suporte

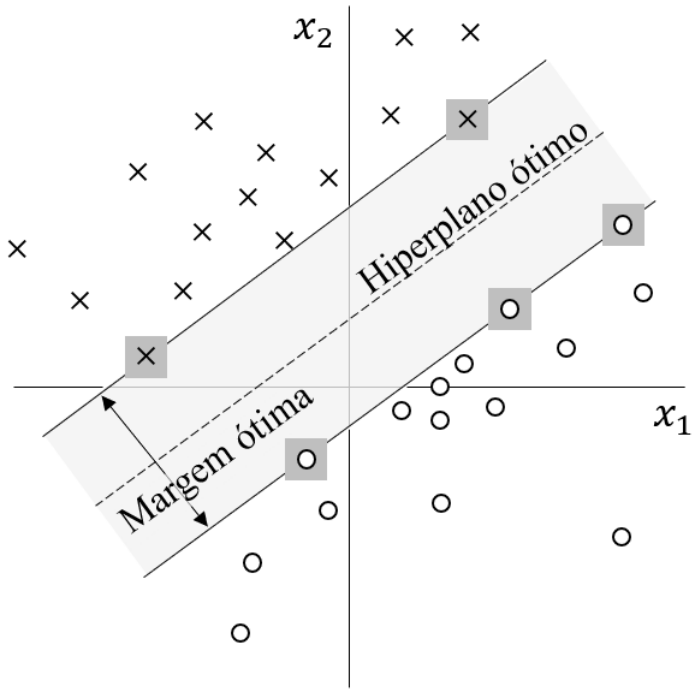
A Máquina de Vetor de Suporte (SVM, *Support Vector Machine*) é uma técnica da aprendizagem de máquina desenvolvida por Vapnik (BOSER; GUYON; VAPNIK, 1992; CORTES; VAPNIK, 1995), e assim como a rede MLP, pode ser utilizada tanto para problemas de classificação quanto de regressão linear.

A Máquina de Vetor de Suporte funciona ao realizar um mapeamento não linear no

conjunto de dados em um espaço de alta dimensão. Nesse espaço uma superfície linear de decisão, denominada de hiperplano, é construída de modo a separar classes distintas (CORTES; VAPNIK, 1995). Desta forma, o hiperplano atua como superfície de decisão de tal forma que a fronteira de separação entre as classes seja máxima (HAYKIN, 2001). A máquina possui esse comportamento baseado na teoria da aprendizagem estatística. Um esquemático de um hiperplano para a separação de um problema bidimensional é mostrado na Figura 15.

Essa seção está dividida em três partes. Na primeira é discutida a construção de um hiperplano linear para problemas com padrões linearmente separáveis, na segunda parte essa restrição é ignorada e será tratada a geração de um hiperplano linear para problemas não linearmente separáveis. A terceira e última parte é dedicada às máquinas de vetor de suporte não lineares.

Figura 15 – Exemplo de um hiperplano para a separação de um problema bidimensional. Os vetores marcados em cinza são chamados de vetores de suporte.



Fonte: Adaptado de Cortes e Vapnik (1995).

A. Máquinas de Vetor de Suporte com margens rígidas

Seja um conjunto de treinamento $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{N}$, com N pontos, onde $\boldsymbol{x}_i$ representa a i-ésima entrada e sua correspondente resposta desejada é dada por d_i. Considere tal conjunto para um problema bidimensional de padrões linearmente separáveis dados pelas

classes $\mathcal{C}_1$, $\mathcal{C}_2$ associadas as saídas $d_i = +1$ e $d_i = -1$, respectivamente. Para este problema, a equação de um hiperplano é dada por:

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \tag{2.35}$$

onde $\boldsymbol{w}$ representa o vetor de pesos ajustáveis e b é um bias. Em especial para um problema bidimensional a Equação 2.35 descreve a equação de uma reta, ou seja, a superfície de decisão é uma reta, como é mostrado na Figura 15, já para um problema tridimensional a superfície é um plano, por exemplo. Uma vez tendo definido a equação do hiperplano, pode-se escrever as seguintes relações:

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}_i + b > 0, \ \boldsymbol{x}_i \in \mathcal{C}_1 \text{ e } d_i = +1 \\ \boldsymbol{w}^T\boldsymbol{x}_i + b < 0, \ \boldsymbol{x}_i \in \mathcal{C}_2 \text{ e } d_i = -1 \end{cases} \tag{2.36}$$

Define-se como margem de separação, ρ, a distância entre o hiperplano e o ponto de dado mais próximo dele. O propósito da Máquina de Vetor de Suporte é encontrar a máxima margem de separação para um dado conjunto de dados. Como ilustrado nas Figuras 15 e 16 denomina-se margem ótima quando a separação é máxima, nessa situação o hiperplano é chamado de hiperplano ótimo (HAYKIN, 2001). Em especial, a margem de uma máquina de vetor de suporte para dados linearmente separáveis é chamada de margem rígida (LORENA; CARVALHO, 2007).

Sejam $\boldsymbol{w}_0$ e b_0 os parâmetros com os quais o hiperplano ótimo é obtido. Então, a Equação 2.35 pode ser reformulada de modo a representar a equação do hiperplano ótimo, como dado a seguir:

$$\boldsymbol{w}_0^T\boldsymbol{x} + b_0 = 0 \tag{2.37}$$

A partir da Equação 2.37 se define a função de decisão D como mostrado na Equação 2.38, a qual fornece uma medida do quão um ponto $\boldsymbol{x}$ está distante do hiperplano.

$$D(\boldsymbol{x}) = \boldsymbol{w}_0^T\boldsymbol{x} + b_0 \tag{2.38}$$

Seja $\boldsymbol{x}_p$ a projeção normal de $\boldsymbol{x}$ sobre o hiperplano ótimo e r a distância algébrica

entre o ponto e o hiperplano, é possível escrever a igualdade vetorial:

$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}_0}{||\boldsymbol{w}_0||} \tag{2.39}$$

Em seguida, como se tem, por definição, que $D(\boldsymbol{x}_p)$ = 0. Então pode-se escrever que:

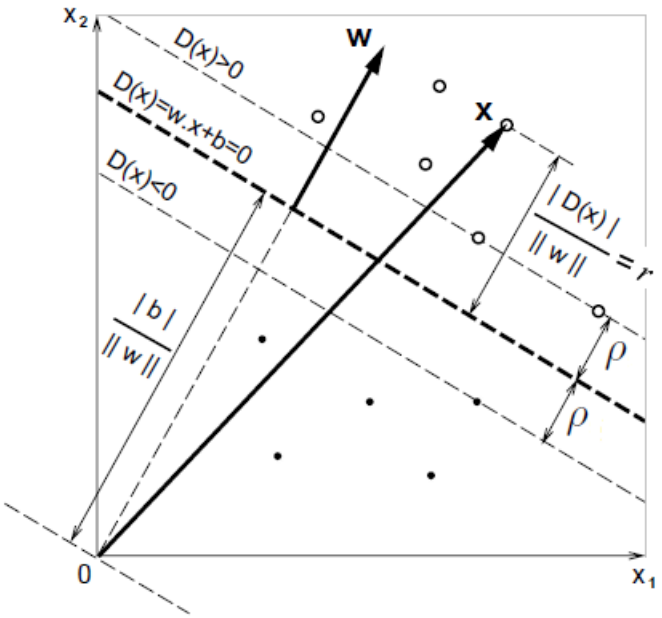
$$D(\boldsymbol{x}) = \boldsymbol{w}_0^T\boldsymbol{x} + b_0 = r||\boldsymbol{w}_0|| \tag{2.40}$$

que reorganizada fica:

$$r = \frac{D(\boldsymbol{x})}{||\boldsymbol{w}_0||} \tag{2.41}$$

As etapas anteriores podem ser melhor entendidas ao analisar a Figura 16. A distância algébrica r pode assumir valores positivos se o ponto $\boldsymbol{x}$ estiver no lado positivo do hiperplano, ou negativo se $\boldsymbol{x}$ estiver no lado negativo (HAYKIN, 2001). Além disso, a distância entre a origem e o hiperplano é dada por $b_0/||\boldsymbol{w}_0||$, pois na origem ($\boldsymbol{x} = \boldsymbol{0}$) tem-se $D(\boldsymbol{x}) = b_0$.

Figura 16 – Interpretação geométrica das distâncias envolvidas entre pontos, a margem de separação e o hiperplano ótimo para um problema bidimensional.



Fonte: Adaptado de Boser, Guyon e Vapnik (1992).

Entende-se como hiperplano canônico quando os pontos dos dados de entradas

mais próximos do hiperplano satisfazem a Equação 2.42 (MULLER et al., 2001).

$$||\boldsymbol{w}_0^T \boldsymbol{x}_i + b_0|| = 1 \tag{2.42}$$

Os pontos que satisfazem a Equação 2.42 é chamado de vetor de suporte. Os vetores de suporte são os vetores mais próximos da superfície de decisão, e portanto, os mais difíceis de classificar (HAYKIN, 2001).

Sejam os vetores de suportes $\boldsymbol{x}_a$ e $\boldsymbol{x}_b$ tais que: $\boldsymbol{w}_0^T \boldsymbol{x}_a + b_0 = 1$ e $\boldsymbol{w}_0^T \boldsymbol{x}_b + b_0 = -1$. Então a margem de separação é dada pela distância entre os vetores $\boldsymbol{x}_a$ e $\boldsymbol{x}_b$ obtida de modo perpendicular ao hiperplano (MULLER et al., 2001). Sendo assim:

$$\begin{aligned} \rho &= (\boldsymbol{x}_a - \boldsymbol{x}_b) \cdot \boldsymbol{w}_0^T / ||\boldsymbol{w}_0|| \\ &= 2/||\boldsymbol{w}_0|| \end{aligned} \tag{2.43}$$

Desta forma, ao analisar a Equação 2.43, entende-se que para encontrar o hiperplano o qual a margem de separação ρ é máxima deve-se minimizar a norma euclidiana do vetor de pesos $\boldsymbol{w}$.

Então, o problema de encontrar um hiperplano ótimo para um conjunto de dados linearmente separáveis é formulado da seguinte maneira:

Dado um conjunto de treinamento $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^N$, deve-se encontrar o vetor de pesos $\boldsymbol{w}$ e o bias b que minimizem a função de custo:

$$\Phi(\boldsymbol{w}) = \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} \tag{2.44}$$

tal que satisfaçam as restrições:

$$d_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 \text{ para } i = 1, 2, \ldots, N \tag{2.45}$$

Vale a pena citar que as restrições expressas na Equação 2.45 é uma reformulação da condição para hiperplano canônico dada na Equação 2.42 considerando também outros vetores que não são de suporte. Ou seja, para vetores de suportes tem-se a igualdade na Equação 2.45 e para os outros tem-se a desigualdade. Além disso, as restrições são impostas de modo a assegurar que não haja nenhum ponto de treinamento entre as margens

de separação das classes, por isso a denominação Máquinas de Vetor de Suporte com margens rígidas (LORENA; CARVALHO, 2007).

O problema de otimização supracitado é chamado de problema primordial, e possui um único mínimo global pois a função $\Phi(\boldsymbol{w})$ é uma função convexa em $\boldsymbol{w}$, e as restrições são lineares em relação a $\boldsymbol{w}$ (HAYKIN, 2001; LORENA; CARVALHO, 2007). O problema primordial pode ser resolvido usando o método dos multiplicadores de Lagrange. O primeiro passo é definir a função de lagrange:

$$J(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{N} \alpha_i \left[d_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\right] \tag{2.46}$$

onde as variáveis não negativas α_i são chamadas de multiplicadores de Lagrange. A função de lagrange deve ser minimizada, o que significa que deve-se maximizar as variáveis α_i e minimizar $\boldsymbol{w}$ e b. Sendo a solução do problema determinado pelo ponto de sela da função de lagrange (HAYKIN, 2001; LORENA; CARVALHO, 2007). Assim, obtêm-se as seguintes condições de otimização:

$$\frac{\partial J(\boldsymbol{w}, b, \alpha)}{\partial \boldsymbol{w}} = \mathbf{0} \tag{2.47}$$

$$\frac{\partial J(\boldsymbol{w}, b, \alpha)}{\partial b} = 0 \tag{2.48}$$

A resolução das condições expressas nas Equações 2.47 e 2.48, resultam em:

$$\sum_{i=1}^{N} \alpha_i d_i = 0 \tag{2.49}$$

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i d_i \boldsymbol{x}_i \tag{2.50}$$

Ao substituir as expressões mostradas pelas Equações 2.49 e 2.50 na função de lagrange 2.46, obtém-se finalmente o problema:

Dado um conjunto de treinamento $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{N}$, deve-se encontrar os multiplicadores

de Lagrange $\{\alpha_i\}_{i=1}^{N}$ que maximizam a função objetivo (HAYKIN, 2001).

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.51}$$

sujeitas às restrições:

1. $\sum_{i=1}^{N} \alpha_i d_i = 0$
2. $\alpha_i \geq 0 \text{ para } i = 1, 2, \ldots, N$

Esta nova formulação do problema é denominada de problema dual e apresenta restrições mais simples além de apresentar o problema de otimização por meio de produtos internos entre pontos de entrada, o que é bastante conveniente para as máquinas de vetor de suporte não lineares (LORENA; CARVALHO, 2007).

B. Construção de hiperplano para padrões não linearmente separáveis

Na seção anterior foi visto como construir um hiperplano para um conjunto de dados com classes linearmente separáveis. Mas, na maioria dos problemas práticos, os conjuntos de dados possuem classes não linearmente separáveis, sendo este caso mais difícil a ser resolvido. Esta seção é dedicada a este tipo de problema.

Quando um ou mais pontos de dados $(\boldsymbol{x}_i, d_i)$ viola a condição expressa na Equação 2.45, denomina-se que a margem de separação entre as classes é suave (HAYKIN, 2001; ABE, 2003).

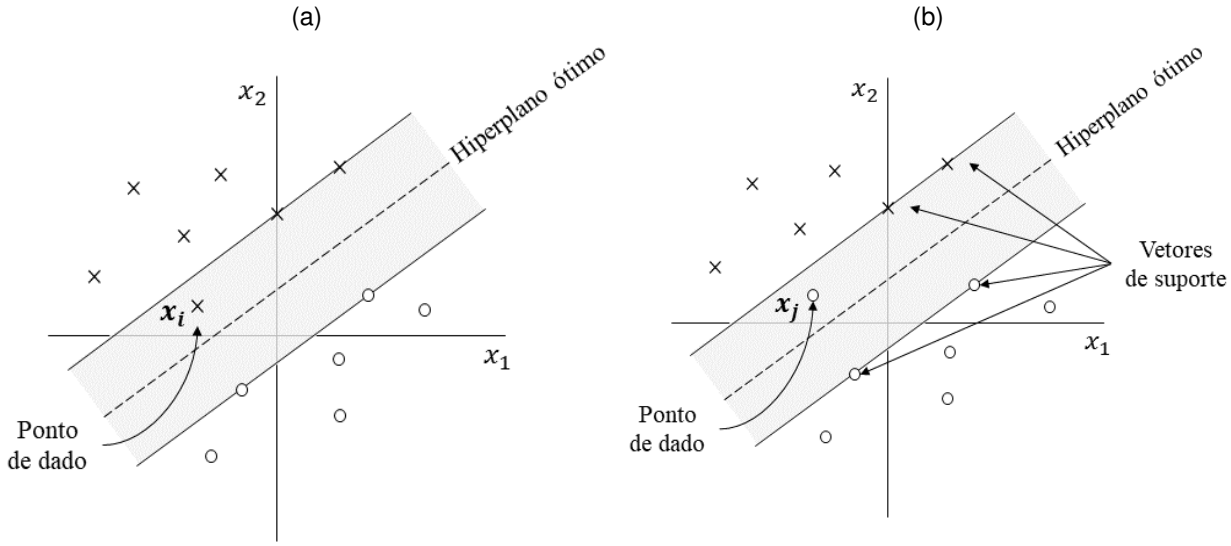
Para esta situação, definem-se as variáveis soltas, representadas por ξ_i. As variáveis soltas são parâmetros não nulos (para $i = 1, 2, \ldots, N$), que medem o desvio do i-ésimo ponto da condição ideal de separabilidade de padrões (HAYKIN, 2001). Isto é feito ao acrescentar as variáveis soltas na definição do hiperplano de separação da seguinte maneira:

$$d_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \;\; i = 1, 2, \ldots, N \tag{2.52}$$

Há duas situações possíveis onde um dado ponto pode violar a condição dada pela Equação 2.45, e em função da sua posição com relação a superfície de decisão se tem

diferentes valores de ξ_i. Para pontos que se encontram dentro da região de separação, mas no lado correto da superfície tem-se $0 \leq \xi_i \leq 1$, situação esta ilustrada pelo ponto $\boldsymbol{x}_i$ na Figura 17a. Enquanto que $\boldsymbol{x}_j$, na Figura 17b, ilustra a situação quando um ponto se encontra do lado errado da superfície, nesse caso se tem $\xi_i > 1$ (HAYKIN, 2001).

Figura 17 – Situações onde a condição de hiperplano rígido é violada: (a) quando o ponto ($\boldsymbol{x}_i$) se encontra do lado correto da superfície de decisão, mas dentro da margem de separação; (b) quando o ponto ($\boldsymbol{x}_j$) se encontra do lado errado da superfície.



Fonte: Adaptado de Haykin (2001).

Nesta nova abordagem, para determinar o hiperplano como superfície de decisão, é necessário encontrar o vetor de peso $\boldsymbol{w}$ e as variáveis soltas ξ_i que minimizam a função de custo $\boldsymbol{\Phi}(\boldsymbol{w}, \xi)$ dada na Eq. 2.53 de modo que $\boldsymbol{w}$ e o bias b satisfaçam a condição da Equação 2.54.

$$\Phi(\boldsymbol{w}, \xi) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{N}\xi_i \tag{2.53}$$

$$\begin{aligned} d_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) &\geq 1 - \xi_i, \text{ para } i = 1, 2, \ldots, N \\ \xi_i &\geq 0, \text{ para } i = 1, 2, \ldots, N \end{aligned} \tag{2.54}$$

onde C é um parâmetro a ser definido pelo usuário que controla a relação entre a maximização da margem suave e a minimização do erro de classificação. Os vetores de suporte serão aqueles vetores que satisfazem a igualdade na Equação 2.54, mesmo se $\xi_i > 0$ (ABE, 2003).

Do mesmo modo para determinar hiperplanos rígidos, a determinação de hiperplanos suaves é feita através do método dos multiplicadores de Lagrange. Seguindo os

mesmos passos descrito na seção anterior obtém-se o problema dual descrito a seguir para padrões não separáveis (HAYKIN, 2001).

Dado um conjunto de treinamento $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^N$, deve-se encontrar os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ que maximizam a função objetivo (HAYKIN, 2001):

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.55}$$

sujeitas às restrições:

1. $\sum_{i=1}^{N} \alpha_i d_i = 0$
2. $0 \leq \alpha_i \leq C \text{ para } i = 1, 2, \ldots, N$

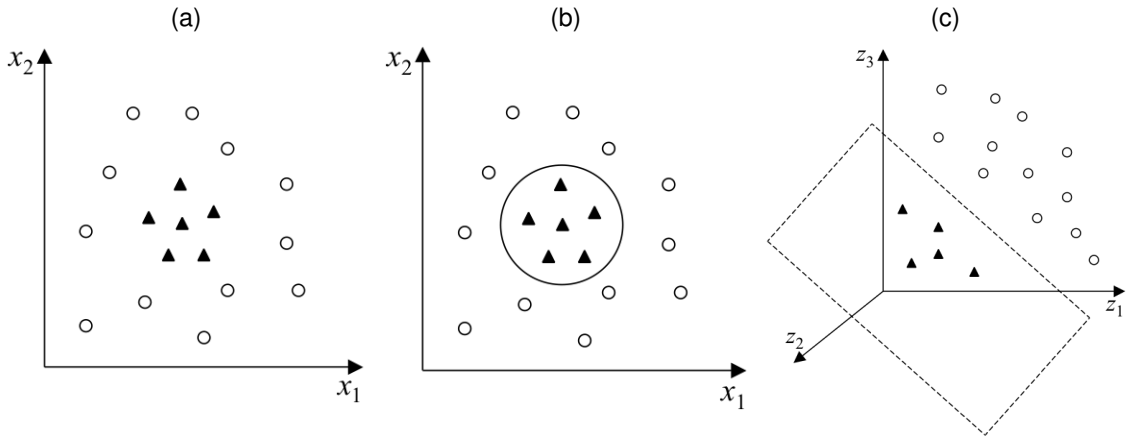
onde C é um parâmetro positivo a ser especificado pelo usuário.

O problema dual para conjuntos não linearmente separáveis é similar ao problema dual para conjuntos linearmente separáveis, exceto pela restrição imposta aos valores das variáveis α_i. No caso não linearmente separável tem-se uma restrição mais rigorosa $0 \leq \alpha_i \leq C$ do que aquela para o caso linearmente separável $\alpha_i \geq 0$. Exceto por esta modificação, a resolução do problema procede do mesmo modo como no caso linearmente separável (HAYKIN, 2001).

C. Máquinas de vetor de suporte não lineares

A construção de um hiperplano suave é eficaz na presença de apenas alguns ruídos e *outliers*. O termo *outliers* se refere a pontos muito distintos dos demais de sua classe, pela presença de ruído em sua obtenção ou por se tratar de casos particulares, raramente presentes no domínio (LORENA; CARVALHO, 2007). Em muitos casos uma superfície de decisão linear não é capaz de dividir de maneira adequada o conjunto de treinamento por um hiperplano (LORENA; CARVALHO, 2007). Na Figura 18a é mostrado uma situação onde uma superfície de decisão linear não é adequada ao conjunto de dados do problema, sendo mais conveniente uma superfície circular como mostrado na Fig. 18b. Nesta seção será discutido como uma máquina de vetor de suporte pode ser usada em situações como essa.

Figura 18 – (a) Conjunto de dados com padrões não linearmente separáveis; (b) Superfície de decisão circular no espaço de entradas; (c) Superfície de decisão linear no espaço de características.



Fonte: Lorena e Carvalho (2007).

De maneira resumida, uma Máquina de Vetor de Suporte mapeia de maneira não linear o conjunto de entrada de treinamento em um espaço de alta dimensão chamado de espaço de características. É, justamente, no espaço de características que é construído o hiperplano ótimo para a separação das classes. Para isto, utiliza-se a versão das máquinas de vetor de suporte com margens suaves, pois esta permite lidar com ruídos presentes nos dados (LORENA; CARVALHO, 2007).

Matematicamente se tem o seguinte: considere o conjunto de treinamento $X \times \mathcal{D}$ onde $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ representa o conjunto dos pontos de entrada e $\mathcal{D} = \{d_i\}_{i=1}^{N}$ é o conjunto das saídas desejadas. O espaço das características $\mathcal{F}$ é então obtido através do mapeamento não linear:

$$\begin{aligned} \varphi : X &\rightarrow \mathcal{F} \\ \boldsymbol{x} &\mapsto \varphi(\boldsymbol{x}) \end{aligned} \tag{2.56}$$

Esse procedimento é feito de acordo com o Teorema de Cover que afirma que para uma transformação $\varphi(\cdot)$ não linear e um espaço de características $\mathcal{F}$ com dimensão suficientemente alta, os padrões serão linearmente separáveis com alta probabilidade. Nesta abordagem, diferente do que vem sendo discutido desde o início desta seção, o hiperplano ótimo será definido como uma função linear de vetores do espaço de características $\mathcal{F}$ e não do espaço das entradas X (HAYKIN, 2001; MULLER et al., 2001; LORENA; CARVALHO,

2007), ou seja, será utilizado como conjunto de aprendizado o conjunto:

$$((\varphi(\boldsymbol{x}_1), d_1), \ldots, (\varphi(\boldsymbol{x}_N), d_N)) \in \mathcal{F} \times \mathcal{D} \tag{2.57}$$

Como um exemplo deste conceito, na Figura 18a é mostrado um conjunto de dados o qual não é linearmente separável em $\mathbb{R}^2$. Contudo, após os pontos de entradas serem mapeados para o $\mathbb{R}^3$ é possível separar as classes com um plano como é mostrado na Figura 18c. Para este caso, o mapeamento não linear é dado na Equação 2.58, e a equação do hiperplano é dado na Equação 2.59 (MULLER et al., 2001; LORENA; CARVALHO, 2007).

$$\begin{aligned} \varphi : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (z_1, z_2, z_3) := \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) \end{aligned} \tag{2.58}$$

$$D(x) = \boldsymbol{w} \cdot \varphi(\boldsymbol{x}) + b = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 = 0 \tag{2.59}$$

Desta maneira o problema dual descrito para as máquinas de vetores de suportes suaves podem ser reformulado considerando agora a imagem da função φ da forma como segue:

Dado um conjunto de treinamento $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^N$, deve-se encontrar os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ que maximizam a função objetivo (HAYKIN, 2001; LORENA; CARVALHO, 2007):

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j (\varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j)) \tag{2.60}$$

sujeitas às restrições:

1. $\sum_{i=1}^{N} \alpha_i d_i = 0$
2. $0 \leq \alpha_i \leq C \text{ para } i = 1, 2, \ldots, N$

onde C é um parâmetro positivo a ser especificado pelo usuário.

O mapeamento φ permite construir uma superfície de decisão não linear no espaço de entrada, mas cuja imagem no espaço de características é linear (HAYKIN, 2001). Contudo, a computação do espaço de características $\mathcal{F}$ pode ser muito custosa ou até inviável em

função da sua alta dimensionalidade. Por outro lado, a única informação necessária do espaço de características é o resultado do produto escalar $\varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)$, para dois pontos $\boldsymbol{x}_i$ e $\boldsymbol{x}_j$, como pode ser visto na Equação 2.60, e isto pode ser obtido através de funções chamadas *kernels* (LORENA; CARVALHO, 2007).

Por definição, um *kernel* é uma função produto interno $K : X \times X \to \mathcal{F}$, isto é, $\forall \boldsymbol{x}_i, \boldsymbol{x}_j \in X$ (HERBRICH, 2001):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) := \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j) \tag{2.61}$$

Por exemplo, o mapeamento φ da Equação 2.58 pelo produto interno de dois vetores $\boldsymbol{x} = (x_1, x_2)$ e $\boldsymbol{y} = (y_1, y_2)$ é dado por (MULLER et al., 2001):

$$\begin{aligned}(\varphi(\boldsymbol{x}) \cdot \varphi(\boldsymbol{y})) &= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) \cdot \left(y_1^2, \sqrt{2}y_1y_2, y_2^2\right)^T \\ &= ((x_1, x_2) \cdot (y_1, y_2)^T)^2 \\ &= (\boldsymbol{x} \cdot \boldsymbol{y})^2 \\ &=: K(\boldsymbol{x}, \boldsymbol{y}).\end{aligned} \tag{2.62}$$

Na prática, utiliza-se a função *kernel* sem ao menos conhecer o mapeamento φ, o qual é gerado implicitamente. Está é a grande vantagem do uso dos *kernels*, devido sua simplicidade em calcular e representar espaços abstratos (LORENA; CARVALHO, 2007).

Para garantir que o *kernel* represente mapeamentos onde seja possível o cálculo de produtos internos, utiliza-se funções *kernel* que seguem as condições estabelecidas pelo teorema de Mercer (HAYKIN, 2001; LORENA; CARVALHO, 2007). Para satisfazer as condições do teorema de Mercer um *kernel* precisar gerar uma matriz positiva semidefinida[5] $\boldsymbol{K}$, em que cada termo K_{ij} é definido por $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\forall i, j = 1, \ldots, N$ (HERBRICH, 2001).

Na Tabela 2 são apresentados alguns tipos de *kernels* comumente utilizados nas Máquinas de Vetor de Suporte, que são: o *kernel* polinomial, a rede de função de base radial e o perceptron de duas camadas. O *kernel* para as Máquinas de Vetor de Suporte do tipo polinomial e função de base radial sempre obedecem o teorema de Mercer, enquanto o *kernel* sigmoidal satisfaz o teorema com algumas restrições como é mostrado na Tabela 2

[5] Seja $\boldsymbol{A}$ uma matriz simétrica $N \times N$ e $\boldsymbol{x}$ um vetor em $\mathbb{R}^N$. Então a matriz $\boldsymbol{A}$ é uma matriz positiva semidefinida se e somente se $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \geq 0$, $\forall \boldsymbol{x} \neq 0$.

(HAYKIN, 2001).

Tabela 2 – Núcleos comumente utilizados nas Máquinas de Vetor de Suporte.

Tipo de Máquina de Vetor de Suporte	Núcleo de produto interno $K(\boldsymbol{x}, \boldsymbol{x}_i)$, $i = 1, 2, \ldots, N$	Observações
Máquina de aprendizagem polinomial	$(\boldsymbol{x}^T \boldsymbol{x}_i + 1)^p$	O parâmetro p deve ser especificado a priori pelo usuário.
Rede de função de base radial	$exp\left(-\frac{1}{2\sigma^2}\|\|\boldsymbol{x} - \boldsymbol{x}_i\|\|^2\right)$	A largura σ^2 deve ser especificada a priori pelo usuário.
Perceptron de duas camadas	$\tanh(\beta_0 \boldsymbol{x}^T \boldsymbol{x}_i + \beta_1)$	O teorema de Mercer é satisfeito apenas para alguns valores de β_0 e β_1.

Fonte: Adaptado de Haykin (2001).

2.4 Redes neurais convolucionais

As Redes Neurais Convolucionais, ou do inglês *Convolutional Neural Networks* (CNNs), são tipos de redes neurais da aprendizagem profunda. O termo aprendizagem profunda ou redes neurais profundas se referem a redes neurais artificiais com muitas camadas (O'SHEA; NASH, 2015; BEZERRA, 2016; ALBAWI; MOHAMMED; AL-ZAWI, 2017). As redes neurais convolucionais possuem esse nome devido à operação matemática linear chamada convolução (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Nas CNNs o dado de entrada é submetido a uma série de convoluções que atuam como extratores de atributos. Ao final das CNNs os atributos extraídos são os dados de entrada de camadas totalmente conectadas que atuam como um classificador.

As CNNs possuem um bom desempenho em problemas de aprendizagem de máquina, como classificação de vídeo (KARPATHY et al., 2014; TRAN et al., 2019), segmentação de imagens (LONG; SHELHAMER; DARRELL, 2015; RONNEBERGER; FISCHER; BROX, 2015; YANG et al., 2020), reconhecimento de voz (LIANG et al., 2017), reconhecimento de modulação (ZHOU; LIU; GRAVELLE, 2020), e por fim, o seu maior domínio de aplicação, o de classificação de imagens (O'SHEA; NASH, 2015; SZEGEDY et al., 2015; SZEGEDY et al., 2016; ALBAWI; MOHAMMED; AL-ZAWI, 2017; CHOLLET, 2017).

Esta seção apresenta conceitos que servem de base para entender as arquiteturas das CNNs. As próximas seções são dedicadas a CNNs utilizadas neste trabalho para a classificação de imagens termográficas, são elas: VGG16, VGG19, ResNet50, MobileNet, Inception V3 e Xception.

Convolução

O processo de convolução em imagens nada mais é do que uma filtragem linear. Neste processo, cada *pixel* da imagem resultante é calculado através de uma média ponderada dos seus *pixels* vizinhos da imagem de entrada (BURGER; BURGE, 2016). A quantidade de vizinhos considerados, ou melhor o tamanho da região de filtragem é um importante parâmetro, pois determina quantos *pixels* contribuirão para cada *pixel* resultante. Esta região é dada pelo tamanho do filtro (também chamado de máscara) que pode assumir, por exemplo, os tamanhos 3×3, 5×5, 7×7, ou até mesmo 21 $\times$ 21, os quais são centrados no *pixel* da imagem de entrada a ser avaliado (BURGER; BURGE, 2016).

Um exemplo de máscara 3×3, H_S, é dada na Equação 2.63. Nela todos os *pixels* são multiplicado por $1/9$ e em seguida somados. O resultado dessa conta será o valor do *pixel* na imagem filtrada que possui posição correspondente ao *pixel* que foi multiplicado pelo elemento central de H_S. O termo $(2, 2)$ em H_S é dedicado justamente para indicar a posição de referência para esta operação. A posição de referência é comumente, mas não necessariamente, utilizada pelo elemento central da máscara (BURGER; BURGE, 2016).

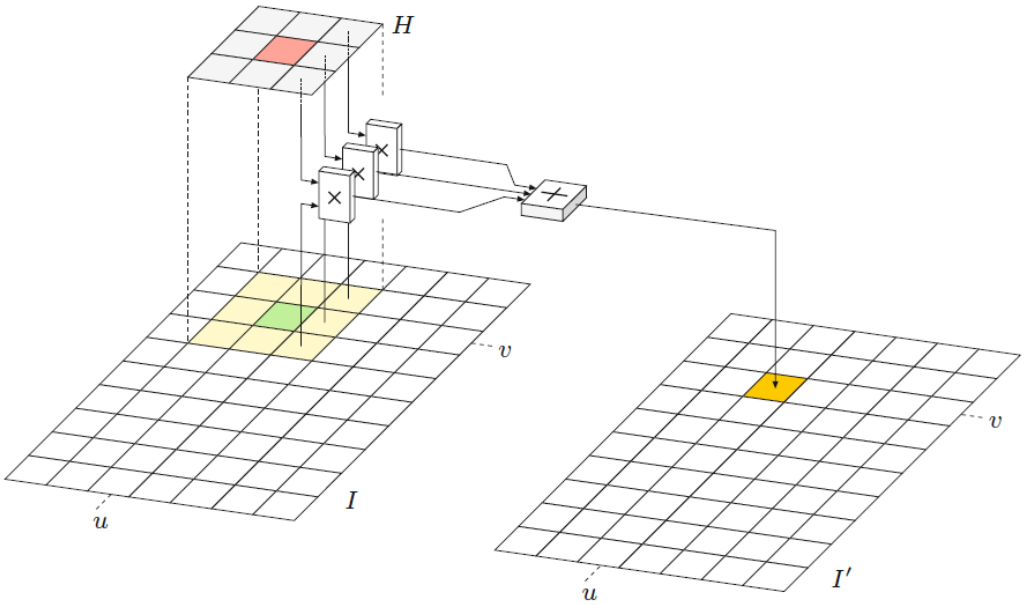
$$H_S = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{(2,2)} \tag{2.63}$$

O filtro dado na Equação 2.63 é um filtro suavizador, pois cada *pixel* da imagem filtrada é obtido através da média aritmética dos seus vizinhos. Nesse processo as quinas e contornos da imagem da entrada são suavizados, borrados, o que dá um aspecto de imagem embaçada após o processo.

A Figura 19 mostra um esquema da realização da convolução. Nesse esquema uma imagem I é convoluída por um filtro H, 3 $\times$ 3, resultando na imagem I'. As coordenadas u e v indicam a posição do *pixel* analisado pelo filtro, de modo que o centro do filtro coincida

com a posição deste *pixel*. Uma vez que todos os coeficientes do filtro são multiplicados pelos *pixels* da região analisada, seus resultados são somados e salvos na posição (u, v) de I'. Matematicamente, o processo de filtragem para uma região R_H é dada pela Equação 2.64 (BURGER; BURGE, 2016).

Figura 19 – Processo de convolução da imagem I com o filtro H. A posição de referência do filtro é posicionada de modo a coincidir com o *pixel* da posição (u, v).

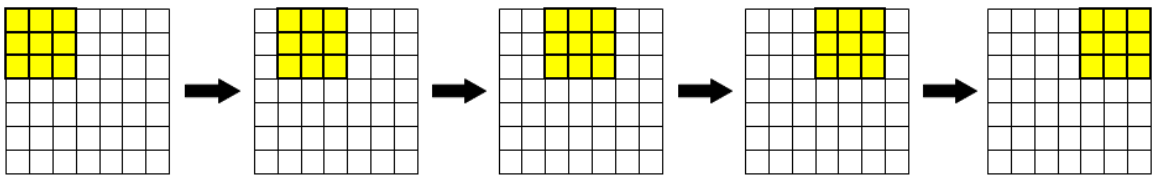


Fonte: Adaptado de Burger e Burge (2016).

$$I'(u,v) = \sum_{(i,j)\in R_H} I(u+i, v+j) \cdot H(i,j) \tag{2.64}$$

Ao final deste processo, a máscara é deslocada de um *pixel* para a direita, e esse processo é repetido até atingir a margem direita da imagem. Em seguida, o filtro desce uma linha e começa todo o processo novamente a partir da margem esquerda da imagem. Isto é feito até que a máscara percorra toda a imagem de entrada. Parte deste processo é mostrado na Figura 20 para uma imagem 7×7 convoluída por um filtro 3×3.

Figura 20 – Durante a convolução a máscara percorre *pixel* a *pixel* a imagem de entrada da esquerda até a direita.



Fonte: O Autor (2022).

A convolução realizada da forma descrita até então é um processo que não avalia os *pixels* das margens da imagem de entrada. Isto pode ser visto na Figura 20, onde os *pixels* da primeira linha da imagem 7×7 não são avaliados pois o filtro 3×3 não pode ser posicionado de modo que a posição de referência do filtro coincida com a posição de algum *pixel* da primeira linha. Para este caso, além dos *pixels* da primeira linha, também não são avaliados os *pixels* da última linha e das margens esquerda e direita. Desta maneira, os *pixels* não avaliados não são salvos na imagem filtrada, o que resulta numa imagem de saída com resolução menor do que a imagem de entrada. Mais especificamente, a imagem resultante da convolução mostrada na Figura 20 é uma imagem 5×5. Uma forma de contornar essa questão é o uso do *padding* que será discutido mais a frente.

Para uma CNN, uma camada convolucional é uma camada da rede responsável por aplicar convoluções de diferentes filtros na imagem de entrada. Além da camada de entrada da rede, outras camadas convolucionais podem ser adicionadas, onde cada camada pode ser associada com filtros diferentes (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

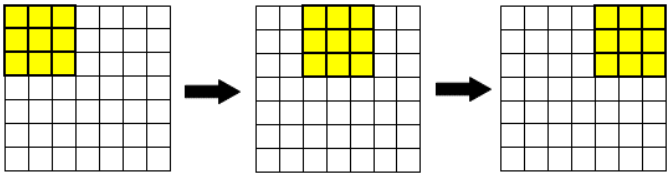
O objetivo das camadas convolucionais em uma CNN é utilizar a convolução como um processo de extração de atributos. Diferente dos métodos de extração de atributos como zernike (KAN; SRINATH, 2001) e haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973) que buscam extrair informações visuais as quais somos capazes de interpretar, como formas e texturas. A extração de atributos via convolução extrai atributos abstratos à medida que a informação de entrada se propaga para camadas mais profundas (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Stride

Ao analisar o esquemático para uma convolução 3×3 em uma imagem 7×7 mostrado na Figura 20 é possível perceber que há uma sobreposição entre as operações. Enquanto a máscara percorre a imagem, um mesmo *pixel* é avaliado diversas vezes. É possível controlar a sobreposição ao configurar um parâmetro chamado *stride* (ALBAWI; MOHAMMED; AL-ZAWI, 2017). A convolução tradicional como explicada anteriormente é realizada com *stride* 1, isto significa dizer que uma vez feita as operações para uma dada região de filtragem a máscara irá se deslocar de um *pixel* para a direita. Ao considerar uma convolução com *stride* 2, por exemplo, a máscara irá se deslocar de dois *pixels* para a

direita, como mostrado na Figura 21. Para *stride* 3 o descolamento é de três *pixels*, e assim por diante por valores maiores de *stride*.

Figura 21 – Esquemático da convolução de uma imagem 7×7 por uma máscara 3×3 com *stride* 2.



Fonte: O Autor (2022).

Como consequência do aumento do *stride* a imagem de saída terá uma resolução menor. Para a convolução com *stride* 2 mostrada na Figura 21 a imagem de saída é uma imagem 3×3. De maneira geral, para uma imagem de resolução $N \times N$ convoluída por uma máscara $F \times F$ com *stride* S resultará em uma imagem de resolução $O \times O$, onde O é dado pela equação (ALBAWI; MOHAMMED; AL-ZAWI, 2017):

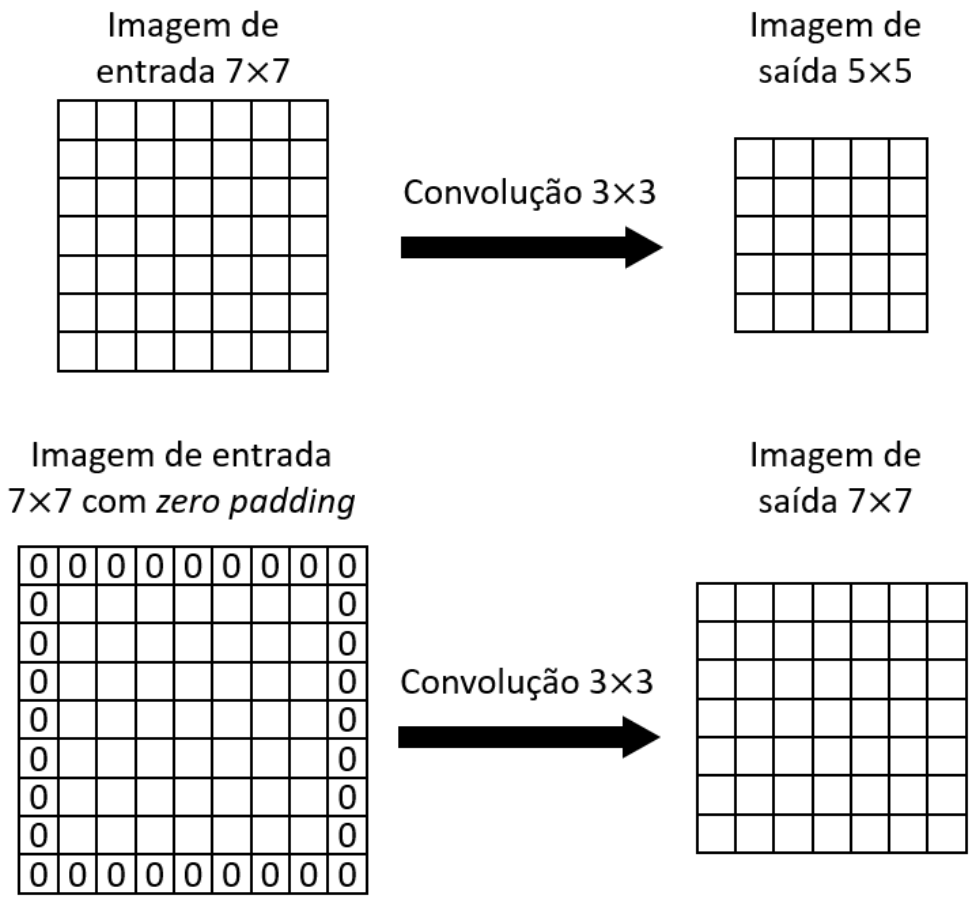
$$O = 1 + \frac{N - F}{S}. \tag{2.65}$$

Padding

Durante a convolução, a informação contida nas bordas da imagem é perdida. Um método simples e eficiente para resolver esse problema é utilizar o preenchimento com zeros (ou do inglês *zero-padding*) (ALBAWI; MOHAMMED; AL-ZAWI, 2017). A ideia do método é simplesmente acrescentar linhas e colunas nas margens da imagem com zeros. Isto é mostrado na Figura 22, onde foi acrescentada uma linha nas margens superior e inferior e uma coluna nas margens esquerda e direita para uma imagem 7×7 convoluída com *stride* 1. O *padding* ilustrado na figura é denominado *padding* 1. Para o *padding* 2 em diante, se adiciona 2 ou mais linhas e colunas em cada margem da imagem. Além de considerar a informação nas bordas, com o *padding* também é possível controlar a resolução da imagem de saída. No processo da Figura 22 é possível perceber que o tamanho da imagem foi preservado durante a convolução com *padding* 1.

A Equação 2.66 dá a relação exata para a resolução da imagem de saída $O \times O$ da convolução de uma imagem $N \times N$, por um filtro $F \times F$, utilizando *stride* de tamanho S e

Figura 22 – Diferença entre uma convolução 3×3 com *stride* 1 sem e com o preenchimento com zeros (*padding*).



Fonte: O Autor (2022).

um *padding* de valor P (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

$$O = 1 + \frac{N + 2P - F}{S} \tag{2.66}$$

Pooling - Downsampling

O objetivo do *pooling* ou *downsampling* é reduzir a resolução da imagem, de modo a reduzir a complexidade para as próximas camadas de uma CNN (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

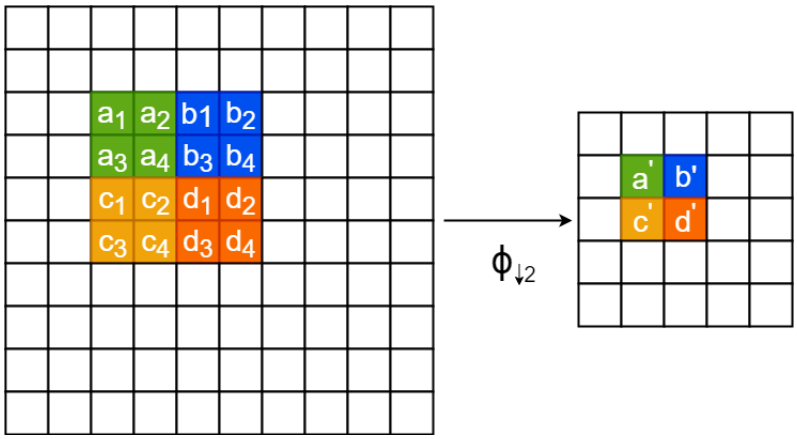
O *pooling* é feito ao substituir quatro *pixels*, disposto em uma janela 2×2, da imagem por apenas um. Outros tamanhos de janelas podem ser considerados, contudo, a janela 2×2 é a mais comumente utilizada (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

A Figura 23 mostra um esquema de como é realizado o *downsampling*. Considere

uma função qualquer $\phi_{\downarrow 2} : \mathbb{R}^4 \rightarrow \mathbb{R}$, onde $\phi_{\downarrow 2}(\cdot)$ pode ser uma função de máximo (retorna o maior valor entre os valores de entrada), ou de mínimo (retorna o menor valor), ou a média aritmética ou mediana do valores do *pixels* ou entre outras possibilidades de funções do tipo $\mathbb{R}^4 \rightarrow \mathbb{R}$. Então, os *pixels* $a^{'}$, $b^{'}$, $c^{'}$, $d^{'}$, identificados na Figura 23, possuem seus valores dados por:

$$a^{'} = \phi_{\downarrow 2}(a_1,\ a_2,\ a_3,\ a_4) \quad b^{'} = \phi_{\downarrow 2}(b_1,\ b_2,\ b_3,\ b_4)$$
$$c^{'} = \phi_{\downarrow 2}(c_1,\ c_2,\ c_3,\ c_4) \quad d^{'} = \phi_{\downarrow 2}(d_1,\ d_2,\ d_3,\ d_4)$$

Figura 23 – *Downsampling* ($\phi_{\downarrow 2}$), através do processo o tamanho da imagem foi reduzido em um quarto.



Fonte: O Autor (2022).

No caso mostrado na Figura 23, a partir de uma imagem 10×10 (100 *pixels*) retorna-se uma imagem 5×5 (25 *pixels*), reduzindo a resolução da imagem por um fator de 4. Também é possível observar na Figura 23 que a janela 2×2 do *pooling* se move pulando dois *pixels*, isso significa que o *pooling* é realizado utilizando um *stride* 2 (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Entre as possíveis opções para a função $\phi_{\downarrow 2}$ a mais comumente utilizada é a função máximo (ALBAWI; MOHAMMED; AL-ZAWI, 2017). O *pooling* quando utiliza a função máximo também é chamado de *max-pooling*, e quando utiliza a função média é chamado de *average-pooling* ou *avg-pooling*.

Batch Normalization

A *batch normalization* (em tradução livre, normalização em lotes) é um método que realiza a normalização das camadas de entradas para cada lote (*batch*) de treinamento da rede (IOFFE; SZEGEDY, 2015). Este processo, visa acelerar o processo de treinamento da

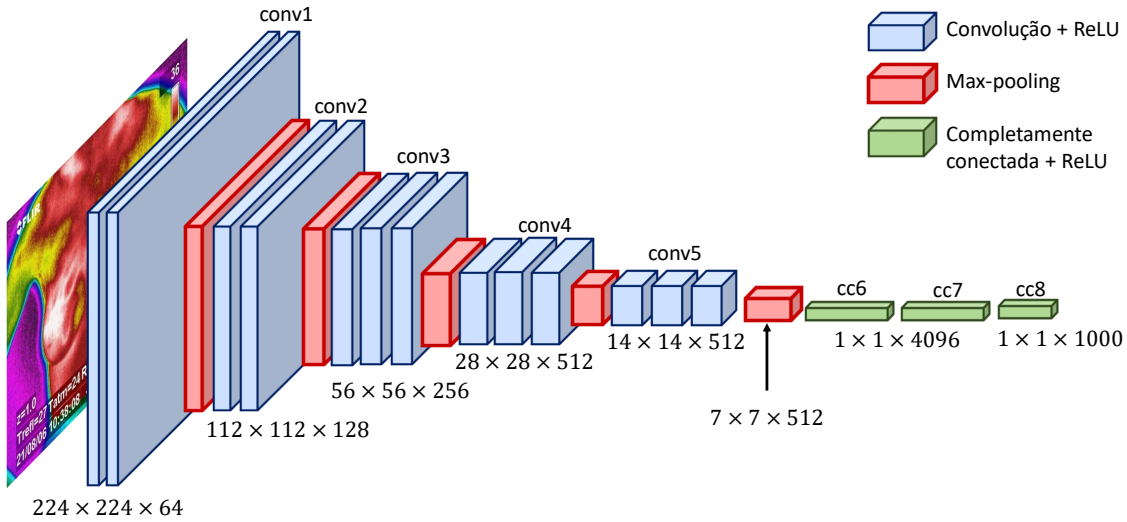
rede ao permitir o uso de maiores taxas de aprendizagem e ser mais robusto a respeito do ponto de inicialização da rede (IOFFE; SZEGEDY, 2015).

Uma das grandes vantagens da *batch normalization* é a incorporação da normalização das funções de ativação na arquitetura da rede. O que garante que a normalização é realizada de forma adequada independente do método de otimização utilizado no treinamento da rede. (IOFFE; SZEGEDY, 2015).

2.4.1 VGGs

As VGGs são CNNs caracterizadas por utilizar, em todas suas camadas convolucionais, máscaras pequenas de tamanho 3×3, com *stride* 1 (SIMONYAN; ZISSERMAN, 2015). As VGG16 e VGG19, dois exemplos de redes VGGs, seguem o mesmo design. Sendo a VGG19 diferente da VGG16 apenas por conter mais camadas convolucionais. Na Tabela 3 são dadas as configurações das duas redes. Na Figura 24 é mostrado um esquemático para a arquitetura da VGG16. Na imagem são dados os tamanhos das máscaras e o número de canais de cada camada convolucional.

Figura 24 – Arquitetura da VGG16. As camadas convolucionais estão representadas pelos blocos em azul e destacadas pelo termo conv. Os blocos em vermelho, com as bordas mais largas, são as camadas *max-pooling*. As três últimas camadas, em verde, são as camadas totalmente conectadas, representadas pelas sigla cc. Em cada camada são mostrados o tamanho da máscara utilizada na convolução e o número de canais.



Fonte: Adaptado de Tabaa et al. (2020).

O treinamento das VGG16 e VGG19 é realizado com imagens RGB de tamanho fixo 224×224. A subtração do valor RGB médio no conjunto de treino para cada pixel é o único pré-processamento feito nas imagens (SIMONYAN; ZISSERMAN, 2015). As camadas

convolucionais das VGG16 e VGG19 utilizam máscaras de tamanho 3×3. O *stride* da convolução é fixo em 1 *pixel*. Enquanto que o *padding* é de 1. Este valor é escolhido de tal maneira que a resolução espacial da imagem seja preservada, desta maneira, para convoluções com máscaras 3×3 é necessário *padding* 1 (SIMONYAN; ZISSERMAN, 2015). O *pooling* espacial é realizado por 5 camadas *max-pooling*, e suas posições de aplicação são mostradas na Tabela 3. Uma janela de *pixels* 2×2, com *stride* 2 é utilizada para a realização do *max-pooling*. Todas as camadas intermediárias usam ReLU como função de ativação (SIMONYAN; ZISSERMAN, 2015).

Por fim, as camadas convolucionais são seguidas por três camadas completamente conectadas, que atuam como um classificador. As duas primeiras camadas possuem 4096 canais, e a última contém 1000 canais. Aqui, o termo canais podem ser entendido da mesma forma do que para imagens, por exemplo, uma imagem RGB possui 3 canais. Por fim, a camada de saída é uma camada que utiliza como função de ativação a função *softmax* (SIMONYAN; ZISSERMAN, 2015). Os dados de saída da função *softmax* podem ser interpretados como probabilidades, dado que esses valores pertencem ao intervalo [0, 1] e sua soma é igual a 1 (BEZERRA, 2016). Desta maneira, o resultado dado pela camada de saída é atrelado a uma certa confiança. Pois, resultados mais confiáveis serão aqueles de maior probabilidade. A VGG16 e a VGG19 possuem 138 e 144 milhões de parâmetros treináveis, respectivamente, como mostrado na Tabela 9.

2.4.2 ResNet

As redes ResNets são baseadas no treinamento residual para facilitar o treinamento de redes profundas (HE et al., 2016). A ideia por trás das ResNets é partir de um modelo de CNN raso (simples) para um modelo mais profundo através da inserção de camadas que realizam um mapeamento identidade.

Como mencionado anteriormente, redes neurais de múltiplas camadas podem ser utilizadas para aproximar qualquer função matemática (CYBENKO, 1988; CYBENKO, 1989). Assim, é correto supor que as redes podem ser capazes de aproximar funções residuais (HE et al., 2016).

Sejam $\boldsymbol{x}$ as entradas da primeira camada de uma pilha de camadas de uma rede (não necessariamente a rede completa) e $\mathcal{H}(\boldsymbol{x})$ o mapeamento a ser aprendido por essas pilhas.

Tabela 3 – Configurações de VGG16 e VGG19. As camadas convolucionais são denotadas por ”conv<tamanho da máscara> - <número de canais>”. As camadas completamente conectadas estão representadas por ”CC - <número de canais>”.

Configuração	
VGG16	VGG19
16 camadas de pesos	19 camadas de pesos
Entrada (imagem RGB 224 $\times$ 244)	
conv3-64 conv3-64	conv3-64 conv3-64
max pool	
conv3-128 conv3-128	conv3-128 conv3-128
max pool	
conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
max pool	
conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
max pool	
conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
max pool	
CC-4096	
CC-4096	
CC-4096	
softmax	

Fonte: Adaptado de Simonyan e Zisserman (2015).

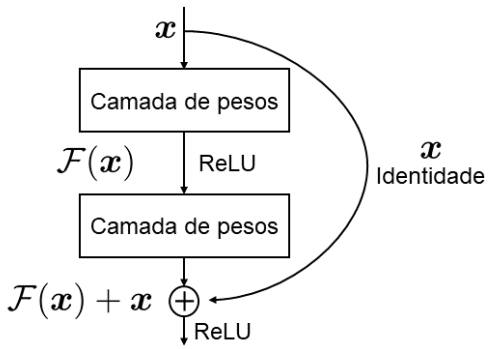
Ao invés de treinar as redes para se aproximar de $\mathcal{H}(\boldsymbol{x})$, o treinamento residual é aplicado para que as redes sejam treinadas para aproximar a função residual $\mathcal{F}(\boldsymbol{x}) := \mathcal{H}(\boldsymbol{x}) - \boldsymbol{x}$ (desde que $\mathcal{H}(\boldsymbol{x})$ e $\boldsymbol{x}$ possuam a mesma dimensão) (HE et al., 2016).

Esta formulação pode ser aplicada para construir um modelo mais profundo a partir

de uma rede rasa, denotada de rede simples. Segundo He et al. (2016), se as camadas acrescentadas forem construídas como mapas de identidade, o modelo profundo deve possuir um erro de treinamento no máximo igual ao seu modelo raso equivalente.

A formulação $\mathcal{F}(\boldsymbol{x}) + \boldsymbol{x}$ é obtida a partir de redes alimentadas adiante através da inserção de conexões de atalho, como mostrado na Figura 25. Esses atalhos pulam uma ou mais camadas realizando um mapeamento identidade, onde sua saída é simplesmente idêntica a entrada (HE et al., 2016).

Figura 25 – Um bloco de construção da aprendizagem residual.



Fonte: Adaptado de He et al. (2016).

Nas ResNets a aprendizagem residual é aplicada a cada algumas pilhas de camadas, formando um bloco de construção, como na Figura 25. Matematicamente, um bloco de construção pode ser modelado por:

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{W_i}) + \boldsymbol{x} \tag{2.67}$$

onde, $\boldsymbol{x}$ e $\boldsymbol{y}$ são as entradas e saídas do bloco, $\mathcal{F}(\boldsymbol{x}, \boldsymbol{W_i})$ representa o mapeamento residual a ser aprendido (HE et al., 2016). No caso da Figura 25, se tem que $\mathcal{F} = \boldsymbol{W_2}\varphi(\boldsymbol{W_1}\boldsymbol{x})$, onde φ representa a função ReLU. A inserção da conexão de atalho na Equação 2.67 não introduz nenhum parâmetro extra. Mas para isso, as dimensões de entrada e saída devem ser a mesma. Caso isso não seja verdade, ou seja, quando há mudança dos canais de entrada e saída, é necessário realizar uma projeção linear $\boldsymbol{W_s}$ nas conexões de atalho para combinar as dimensões (HE et al., 2016):

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{W_i}) + \boldsymbol{W_s}\boldsymbol{x}. \tag{2.68}$$

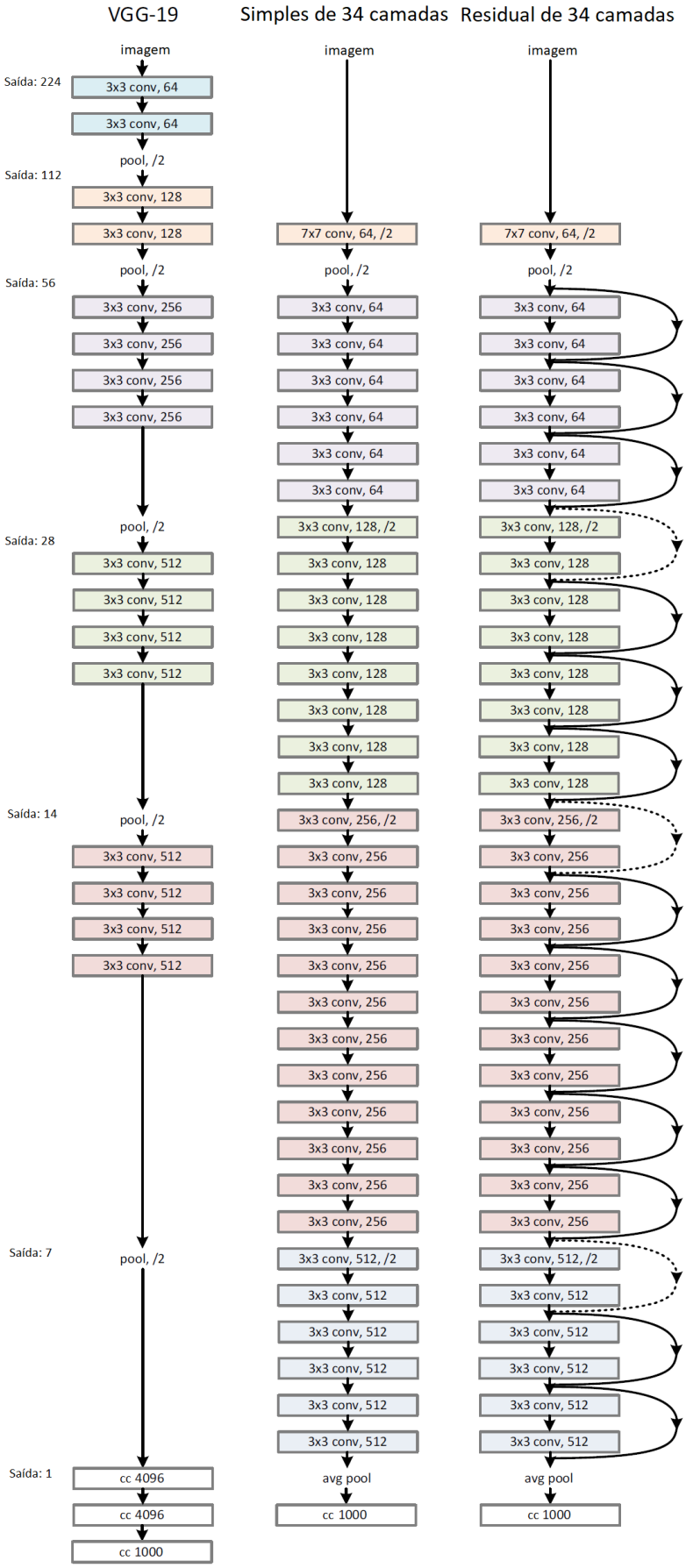
A arquitetura da rede simples foi inspirada nas redes VGGs. Na sua maioria, as

redes convolucionais possuem máscaras de tamanho 3 $\times$ 3 e seguem duas regras: 1) se o mapeamento de atributos de saída possui a mesma dimensão da entrada, então as camadas possuem o mesmo número de filtros; 2) se o tamanho do mapeamento é dividido pela metade, o número de filtros é dobrado de modo a preservar o tempo computacional por camada (HE et al., 2016). Além disso, o *downsampling* é executado através das camadas convolucionais que possuem *stride* de 2. Por fim, a rede simples executa um *pooling* da média e em seguida se tem uma camada completamente conectada de 1000 canais com *softmax* (HE et al., 2016).

A Figura 26 mostra a arquitetura da VGG19, a arquitetura de uma rede simples contendo 34 camadas de pesos, e a rede residual equivalente de 34 camadas, denotada de ResNet34. A rede residual é obtida através da inserção das conexões de atalhos na sua rede simples equivalente. Os atalhos de identidade são usados diretamente quando a entrada e a saída possuem as mesmas dimensões (HE et al., 2016). Esses atalhos são representados pelas linhas sólidas na Figura 26, já as linhas tracejadas, representam os atalhos quando a dimensão aumenta. Neste caso, são consideradas duas opções: 1) o mapeamento identidade é executado, mas nesse caso, as entradas são acrescidas de zeros através do *padding*, para que a dimensão de entrada seja igual a da saída. Nessa opção, nenhum parâmetro é acrescentado à rede; 2) O atalho de projeção da Equação 2.68 é utilizado para igualar as dimensões (através de uma convolução 1$\times$1). Nas duas situações anteriores, os atalhos são realizados com *stride* 2 (HE et al., 2016).

A Tabela 4 mostra as arquiteturas da ResNet com 50 camadas (ResNet50), e da ResNet34. Na tabela, os blocos de construção são dados dentro de colchetes, seguidos do número de blocos colocado em sequência. O *downsampling* é executado por conv3_1, conv4_1, e conv5_1 com *stride* 2 (HE et al., 2016). A ResNet50 é obtida ao substituir cada bloco de construção de duas camadas da ResNet34 por um bloco com três camadas. Esse novo bloco possui camadas com convoluções de máscaras 1$\times$1, 3$\times$3 e 1$\times$1 (HE et al., 2016). As camadas 1$\times$1 são responsáveis por diminuir e depois aumentar a dimensão, o que deixa a camada 3$\times$3 com menores dimensões de entrada e saída, isto é importante para a diminuição do custo computacional do modelo (HE et al., 2016). Essa abordagem é chamada de gargalo. Assim como as VGG16 e VGG19, as ResNets trabalham por padrão com imagens de entrada com resolução 224$\times$224 (HE et al., 2016). Ao todo, a ResNet50 possui 25,6 milhões de parâmetros treináveis, como na Tabela 9.

Figura 26 – Exemplo de arquitetura da ResNet com 34 camadas de parâmetros. À esquerda, VGG19, no centro, a rede simples, à direita, a rede residual (ResNet34). Em destaque à esquerda estão as dimensões de saída para a VGG19.



Fonte: Adaptado de He et al. (2016).

Tabela 4 – Arquiteturas das ResNet34 e ResNet50. Em colchetes são dados os blocos de construção, seguidos do número de blocos colocado em sequência. O *downsampling* é executado por conv3_1, conv4_1, e conv5_1 com *stride* 2.

Nome da camada	Tamanho da saída	34-camadas	50-camadas
conv1	112 x 112	7×7, 64 canais, stride 2	
conv2_x	56x56	3×3 max pool, stride 2	
		$\begin{bmatrix} 3\times3,\ 64 \\ 3\times3,\ 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 64 \\ 3\times3,\ 64 \\ 1\times1,\ 256 \end{bmatrix}\times3$
conv3_x	28x28	$\begin{bmatrix} 3\times3,\ 128 \\ 3\times3,\ 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1,\ 128 \\ 3\times3,\ 128 \\ 1\times1,\ 512 \end{bmatrix}\times4$
conv4_x	14x14	$\begin{bmatrix} 3\times3,\ 256 \\ 3\times3,\ 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1,\ 256 \\ 3\times3,\ 256 \\ 1\times1,\ 1024 \end{bmatrix}\times6$
conv5_x	7x7	$\begin{bmatrix} 3\times3,\ 512 \\ 3\times3,\ 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1,\ 512 \\ 3\times3,\ 512 \\ 1\times1,\ 2048 \end{bmatrix}\times3$
	1x1	average pool, CC-1000, softmax	

Fonte: He et al. (2016).

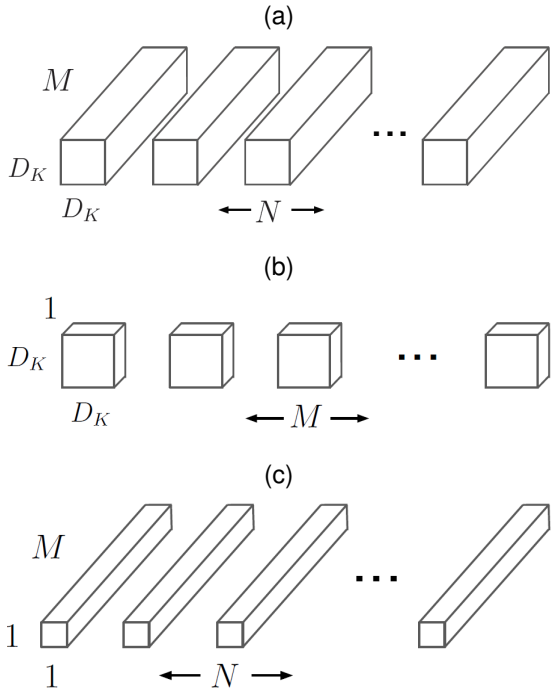
2.4.3 MobileNet

A MobileNet é baseada na convolução separável em profundidade (*Depthwise Separable Convolution*) (HOWARD et al., 2017). Nela os filtros convolucionais tradicionais são substituídos por uma combinação dos filtros convolucionais em profundidade (*Depthwise Convolutional Filters*) com filtros convolucionais pontuais (*Pointwise Convolutional Filters*), a qual se trata de uma convolução 1×1. A união destes filtros recebe o nome de filtros separáveis em profundidade.

Segundo Chollet (2017), a convolução separável em profundidade consiste em uma convolução espacial realizada de forma independente para cada canal de uma entrada, seguida da convolução pontual para projetar os canais de saída da convolução em profundidade em um novo espaço de canais. Isto é, os filtros convolucionais em profundidade são aplicados para cada canal de entrada, e suas saídas são combinadas através da convolução pontual. Na convolução tradicional estas tarefas são realizadas em uma única etapa, a separação das tarefas em duas etapas na convolução separável em profundidade reflete

numa redução substancial do custo computacional do modelo (HOWARD et al., 2017). A MobileNet utiliza ReLU e *batch normalization* tanto para a camada responsável pela convolução em profundidade, quanto para a camada responsável pela convolução pontual. A Figura 27 mostra um esquema para a convolução tradicional, a convolução em profundidade e a convolução pontual, onde D_K representa o comprimento e a largura do mapeamento de entrada, M e N são os números de canais da entrada e saída, respectivamente.

Figura 27 – Esquemático da (a) convolução tradicional, (b) convolução em profundidade e (c) convolução pontual.



Fonte: Howard et al. (2017).

A arquitetura da MobileNet é dada na Tabela 5. A primeira camada da MobileNet é uma camada convolucional completa. Todas as outras camadas convolucionais executam a convolução separável em profundidade. As camadas responsáveis pela convoluções em profundidade estão identificados na Tabela 5 pela sigla dw para *depthwise*. O *downsampling* é realizado pelas camadas convolucionais com o uso do *stride*. O valor do *stride* de cada camada é dado na primeira coluna da Tabela 5. Após uma pilha de camadas convolucionais a MobileNet realiza um *pooling* da média para reduzir a resolução espacial para um. A rede termina com uma camada completamente conectada de 1000 canais e uma camada *softmax*. Considerando as convoluções em profundidade e pontual como camadas diferentes, ao total, a MobileNet possui 28 camadas (HOWARD et al., 2017).

A arquitetura discutida até então é referente a MobileNet padrão. Mas o modelo

Tabela 5 – Arquitetura da MobileNet

Tipo / Stride	Tamanho do Filtro	Dimensão de Entrada
Conv / s2	3×3×3×32	224×224×3
Conv dw / s1	3×3×32 dw	112×112×32
Conv / s1	1×1×32×64	112×112×32
Conv dw / s2	3×3×64 dw	112×112×64
Conv / s1	1×1×64×128	56×56×64
Conv dw / s1	3×3×128 dw	56×56×128
Conv / s1	1×1×128×128	56×56×128
Conv dw / s2	3×3×128 dw	56×56×128
Conv / s1	1×1×128×256	28×28×128
Conv dw / s1	3×3×256 dw	28×28×256
Conv / s1	1×1×256×256	28×28×256
Conv dw / s2	3×3×256 dw	28×28×256
Conv / s1	1×1×256×512	14×14×256
5× Conv dw / s1 Conv / s1	3×3×512 dw 1×1×512×512	14×14×512 14×14×512
Conv dw / s2	3×3×512 dw	14×14×512
Conv / s1	1×1×512×1024	7×7×512
Conv dw / s2	3×3×1024 dw	7×7×1024
Conv / s1	1×1×1024×1024	7×7×1024
Avg Pool / s1	Pool 7×7	7×7×1024
CC / s1	1024×1000	1×1×1024
Softmax / s1	Classificador	1×1×1000

Fonte: Adaptado de Howard et al. (2017).

também prevê a utilização de dois hiperparâmetros que permite ao modelo se moldar a aplicação de modo a diminuir o custo computacional do processo. O primeiro hiperparâmetro é o Multiplicador de Largura, o qual é responsável por afinar as camadas da rede, isto é, diminuir os canais de entrada e saída de cada camada. O segundo hiperparâmetro é o Multiplicador de Resolução, aplicável na imagem de entrada, reduzindo sua resolução. A MobileNet trabalha com imagens de resoluções 224, 192, 160 ou 128, sendo 224 a resolução padrão (HOWARD et al., 2017).

Neste trabalho, a versão da MobileNet utilizada foi a versão padrão, sem a utilização

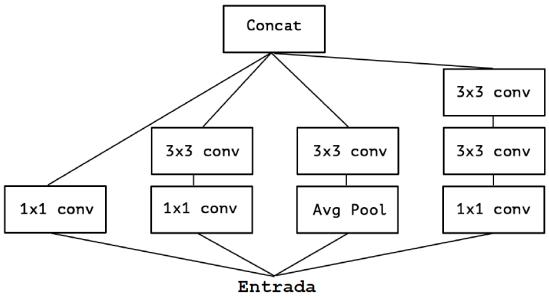
dos multiplicadores. O número total de parâmetros treináveis da rede, dada na Tabela 9, é de 4,2 milhões.

2.4.4 Inception V3

As redes Inceptions foram primeiramente propostas por SZEGEDY et al. em 2014, para o desafio *ImageNet Large-Scale Visual Recognition Challenge 2014* (ILSVRC14). Esta rede foi chamada de Inception V1 ou GoogLeNet. Posteriormente, a GoogLeNet foi aprimorada para as versões Inception V2 e Inception V3 (SZEGEDY et al., 2016). Após essas, outras versões vieram, como a Inception V4 (SZEGEDY et al., 2017), Inception-ResNet, a qual é resultado da combinação da ResNet (HE et al., 2016) com a Inception V3 (SZEGEDY et al., 2017), e Xception, versão da Inception que faz uso das convoluções separáveis em profundidade (CHOLLET, 2017).

O bloco fundamental das redes Inception é dado na Figura 28. Este bloco é considerado como bloco canônico, pois muitas versões da Inception são baseadas nele. No bloco Inception o dado de entrada é submetido a quatro ramos de procedimentos diferentes e suas saídas são concatenadas. Um modelo Inception é construído a partir de uma pilha de blocos como esse (CHOLLET, 2017).

Figura 28 – Bloco canônico das redes Inceptions.



Fonte: Adaptado de Chollet (2017).

Esta seção é dedicada a apresentar a arquitetura da Inception V3. Para isto, conceitos e procedimentos fundamentais desta rede serão apresentados primeiro.

Fatorização em convoluções menores

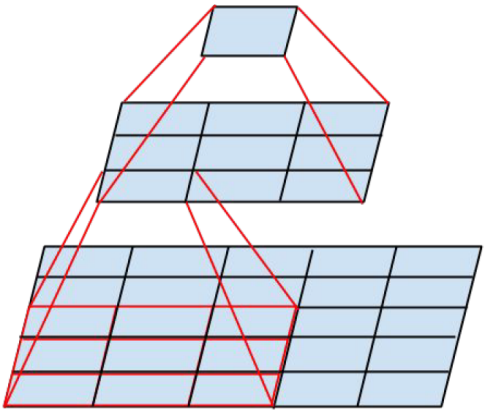
As redes Inceptions são completamente convolucionais, desta maneira, cada peso corresponde a uma multiplicação por ativação. Assim, qualquer redução no custo computa-

cional resulta na redução no número de parâmetros da rede (SZEGEDY et al., 2016).

A arquitetura da Inception V1 possui convoluções com grandes filtros espaciais 5×5 e 7×7 (SZEGEDY et al., 2015). Contudo, tais filtros tendem a ser bastante custosos computacionalmente (SZEGEDY et al., 2016). Logo, para a Inception V3 é proposto um meio de fatorar grandes convoluções em convoluções menores de modo a aumentar a eficiência computacional da rede, diminuindo-se a quantidade total de parâmetros, mas mantendo o mesmo tamanho da entrada e a profundidade de saída.

Na proposta da Inception V3, um filtro 5×5 pode ser substituído por uma arquitetura de duas camadas convolucionais de 3×3 (SZEGEDY et al., 2016), como é esquematizado na Figura 29. Essa configuração é equivalente a convolução 5×5 em termos de expressividade, ao passo que tem menor quantidade de parâmetros, pois os pesos entre regiões adjacentes são compartilhados. Para a fatorização do filtro 5×5 a redução do custo computacional é de 28 % (SZEGEDY et al., 2016).

Figura 29 – Fatorização de uma convolução 5×5 em uma pequena rede convolucional de duas camadas com filtros 3×3.



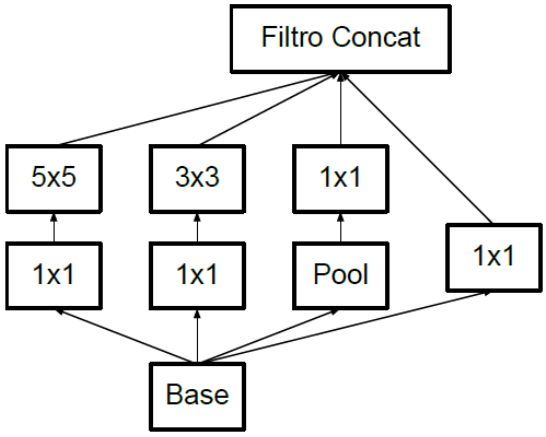
Fonte: Szegedy et al. (2016).

A Figura 30 mostra o módulo Inception original proposta para a Inception V1. Com a fatorização em convoluções menores, a convolução 5×5 presente nesse módulo é substituída por duas convoluções 3×3. Essa mudança pode ser percebida no módulo Inception utilizado na versão 3 mostrado na Figura 31.

Fatorização espacial com convoluções assimétricas

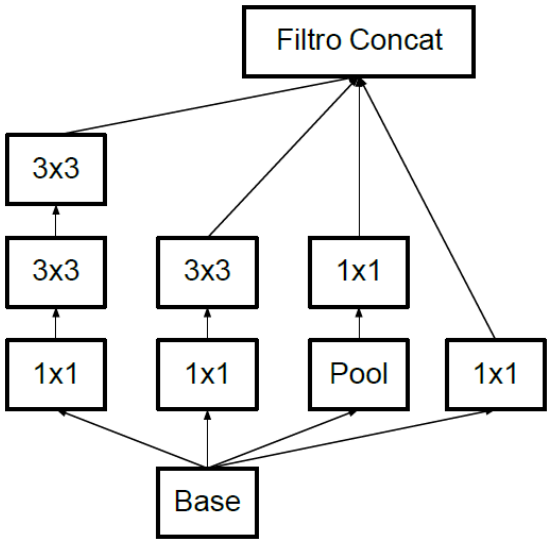
A Inception V3 também possui um outro tipo de fatorização baseado em convoluções assimétricas. Nesta proposta, uma convolução $n \times n$ é substituída por uma convolução

Figura 30 – Módulo Inception, da Inception V1. Destaque para a convolução 5×5 presente no ramo esquerdo.



Fonte: Szegedy et al. (2016).

Figura 31 – Módulo Inception, da Inception V3, onde uma convolução 5×5 é substituída por duas convoluções 3×3.



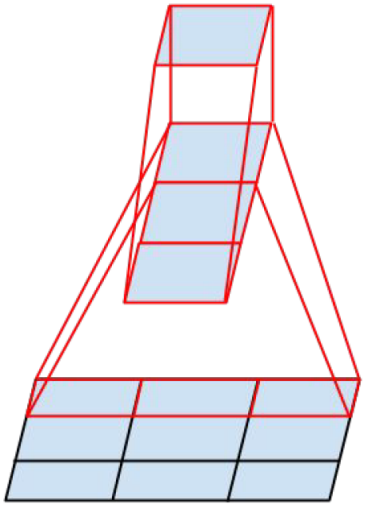
Fonte: Szegedy et al. (2016).

$1 \times n$ seguida de uma convolução $n \times 1$ (SZEGEDY et al., 2016), por exemplo, na Figura 32 é exemplificada a fatorização de um filtro 3×3 por um filtro 1×3 seguido por um filtro 3×1.

Na fatoração assimétrica, quanto maior for o valor de n maior será a redução do custo computacional (SZEGEDY et al., 2016). Segundo Szegedy et al. (2016), esta fatorização entrega bons resultados para resoluções de tamanho médio, isto é, para um mapeamento $m \times m$, onde m varia de 12 até 20. Em especial, a fatorização assimétrica pode encontrar bons resultados ao utilizar uma convolução 1×7 seguida por outra convolução 7×1.

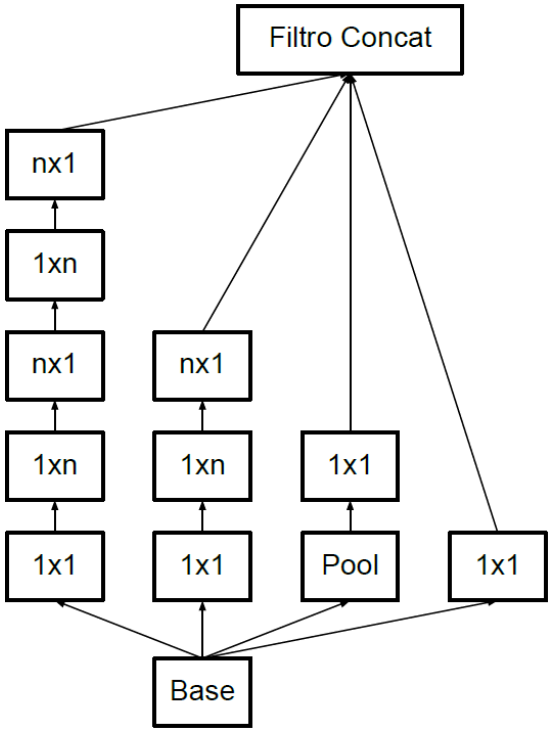
Na Figura 33 é mostrado o módulo Inception onde uma convolução $n \times n$ é fatorada em uma convolução $1 \times n$ seguida por outra de $n \times 1$. Na arquitetura da Inception V3 é utilizado $n = 7$ para uma resolução 17×17.

Figura 32 – Fatorização assimétrica de uma convolução 3×3 por uma convolução 1×3 seguida de outra 3×1.



Fonte: Szegedy et al. (2016).

Figura 33 – Módulo Inception com convoluções $n \times n$ fatoradas assimetricamente.



Fonte: Szegedy et al. (2016).

Redução da resolução

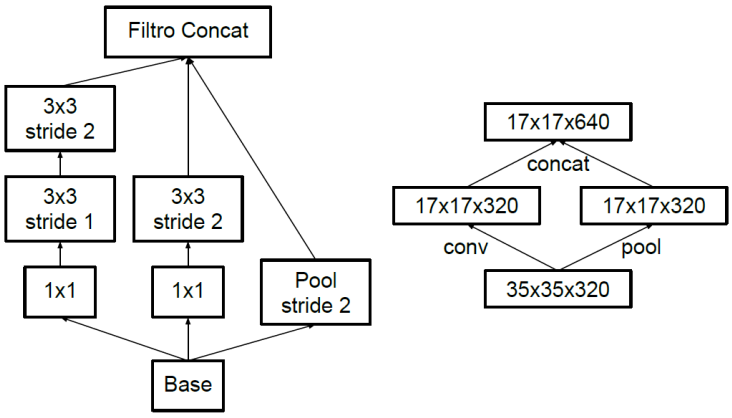
Tipicamente, uma CNN utiliza uma camada de *pooling* para a redução da dimensionalidade dos dados. Contudo esse processo pode não ser tão eficiente.

Considerando um dado de entrada de dimensão $d \times d$ com k canais. Se for desejado passar esses dados para a dimensão $\frac{d}{2} \times \frac{d}{2}$ com $2k$ canais. Então é necessário submeter o

dado de entrada a uma convolução com $2k$ filtros, com *stride* 1, e em seguida, realizar um *pooling*. Isso faz com que boa parte do custo computacional seja realizado por custosas convoluções em uma grande resolução utilizando $2d^2k^2$ operações.

Uma alternativa para diminuir o custo computacional desta operação é utilizar dois blocos paralelos, como mostrado na Figura 34. O primeiro bloco é uma camada convolutiva, este bloco é representado pelas duas colunas da esquerda na Figura 34. Já o segundo é uma camada de *pooling* (pode ser tanto o *max-pooling*, quanto *avg-pooling*), sendo os dois realizados com *stride* 2. As saídas dos dois blocos são, enfim, concatenadas (SZEGEDY et al., 2016). Na Figura 34, o diagrama da esquerda representa as operações, sendo as duas colunas da esquerda responsáveis pela convolução, e a coluna da direita responsável pelo *pooling*. Na direita, está o diagrama indicando a redução de resolução durante o processo.

Figura 34 – Módulo de redução da resolução através da filtragem.



Fonte: Szegedy et al. (2016).

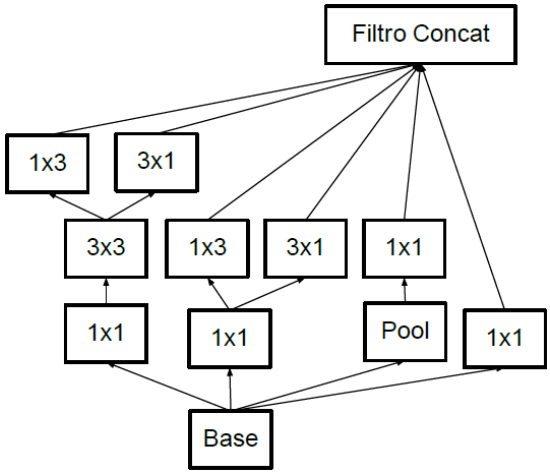
Na Figura 35 é mostrado o módulo Inception com a redução do tamanho da resolução. Esse módulo é utilizado para resoluções de tamanho 8×8 para promover uma representação de alta dimensão (SZEGEDY et al., 2016), sendo assim, o módulo da Figura 35 é utilizado apenas nas últimas camadas convolucionais da Inception V3.

Arquitetura

A Inception V3 é obtida a partir da Inception V2 com apenas algumas mudanças. Desta forma, será apresentada a arquitetura da Inception V2 e por fim indicado quais mudanças realizadas para a terceira versão.

A arquitetura da Inception V2 está discriminada na Tabela 6. A primeira camada

Figura 35 – Módulo Inception com a redução do tamanho da resolução.



Fonte: Szegedy et al. (2016).

da Inception V1 é uma camada convolucional 7×7 (SZEGEDY et al., 2015), na Inception V2 esta camada é fatorada em três convoluções 3×3 (SZEGEDY et al., 2016), mostradas nas três primeiras linhas da tabela. A parte Inception da CNN é formada por três módulos tradicionais Inception de 35×35 com 288 canais cada, como mostrado na Figura 31. Esses dados são reduzidos para 17×17 com 768 canais utilizando a redução de resolução. E em seguida, há 5 módulos com fatoração assimétrica (Figura 33). Novamente os dados têm sua dimensionalidade reduzida, agora para 8×8×1280, e são submetidos a dois módulos como o mostrado na Figura 35 (SZEGEDY et al., 2016). A camada que utiliza *zero-padding* está marcada na tabela, esse procedimento é utilizado para manter a resolução dos dados. Além disso, o *padding* é utilizado dentro dos módulos Inceptions que não reduzem a dimensão dos dados. Exceto essas, todas as outras camadas não usam o *padding*. Ao todo, a Inception V2 possui 42 camadas (SZEGEDY et al., 2016).

A partir da Inception V2 se obtém a Inception V3 ao acrescentar ao modelo os seguintes procedimentos: *Label Smoothing* e classificadores auxiliares com *batch normalization*. O *Label Smoothing* é um mecanismo para regularizar o classificador (SZEGEDY et al., 2016). Os classificadores auxiliares são camadas completamente conectadas que obtêm como dados de entrada resultados parciais da CNN. Sendo assim, o fluxo de informação que segue do dado de entrada até o classificador principal da rede possui um ou mais ramos para o classificadores auxiliares. Para a Inception V3 a motivação do uso de tais classificadores normalizados é utilizá-los como regularizadores (SZEGEDY et al., 2016). Na Figura 36 é dado um esquemático de um classificador auxiliar posicionado na saída da

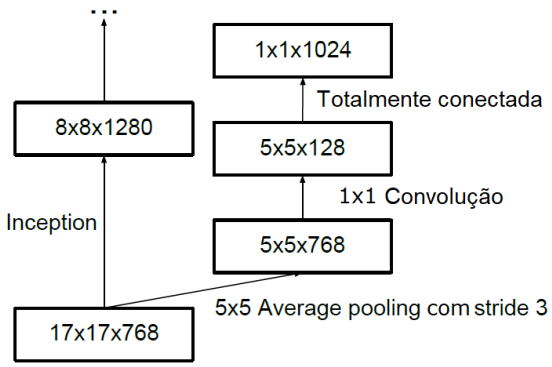
Tabela 6 – Arquitetura da Inception V2

Tipo	Tamanho do filtro/*stride*	Tamanho da Entrada
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv (*padding*)	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	Como na Figura 31	35×35×288
5×Inception	Como na Figura 33	17×17×768
2×Inception	Como na Figura 35	8×8×1280
pool	8×8	8×8×2048
linear	logits	1×1×2048
softmax	Classificador	1×1×1000

Fontes: Szegedy et al. (2016).

última cada 17×17.

Figura 36 – Classificador auxiliar



Fonte: Adaptado de Szegedy et al. (2016).

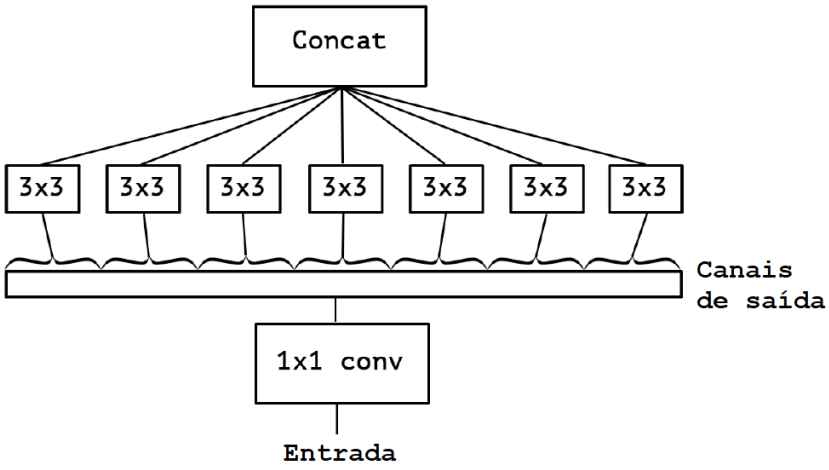
2.4.5 Xception

A versão extrema das redes Inceptions é conhecida como Xception (contração da expressão em inglês *"Extreme Inception"*), nela os módulos Inception (Figura 28) são substituídos por convoluções separáveis em profundidade (CHOLLET, 2017).

Na Xception o mapeamento das correlações entre canais e o mapeamento das correlações espaciais são realizados de uma maneira totalmente independente e separada. Desta maneira, o módulo Xception, mostrado na Figura 37, realiza primeiro uma convolução 1×1 para mapear correlações entre canais, e em seguida são realizadas convoluções 3×3 para mapear separadamente correlações espaciais para cada canal de saída. Esta

operação é similar a convolução separável em profundidade (CHOLLET, 2017). A única diferença entre o módulo extremo da Inception e a convolução separável em profundidade realização pela MobileNet é a ordem das operações. Na MobiletNet primeiro é realizado a convolução 3×3, a chamada convolução em profundidade. E em seguida, é realizada a convolução 1×1, a qual é a convolução pontual. Na Xception essa ordem é invertida.

Figura 37 – Módulo Xception.

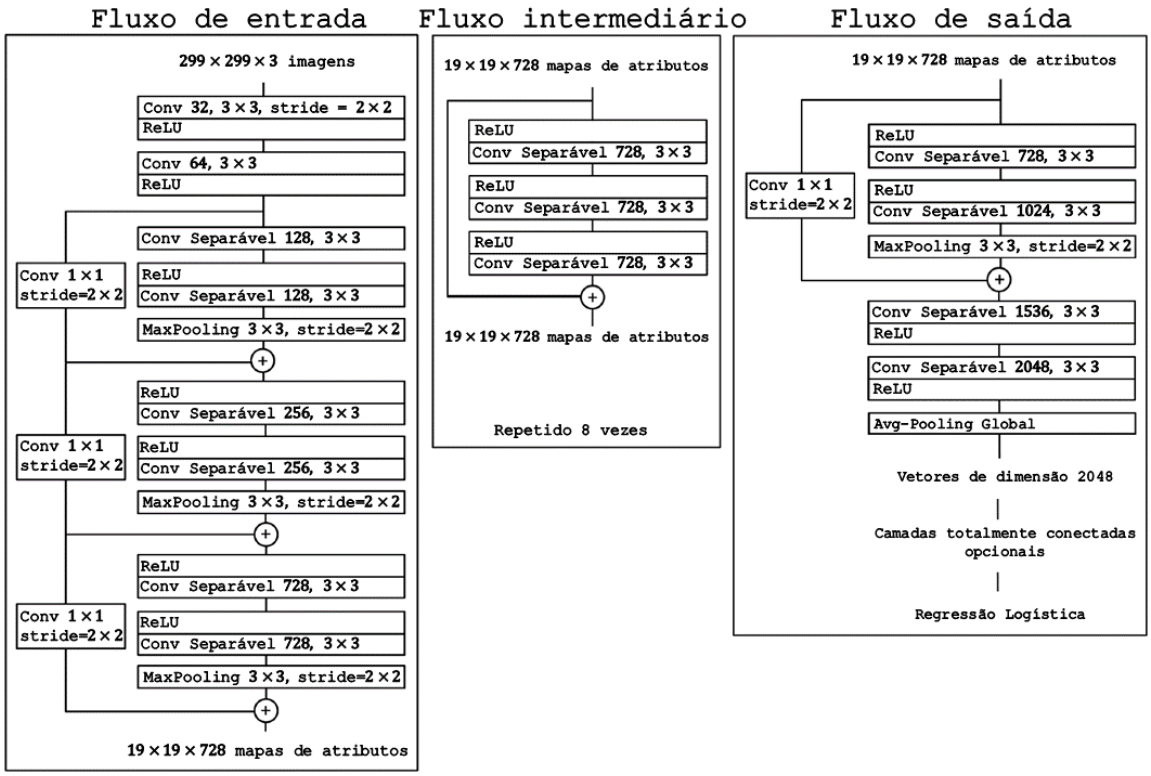


Fonte: Adaptado de Chollet (2017).

Na Figura 38 é mostrada a arquitetura da Xception, os dados de entrada seguem o fluxo das colunas de entrada, intermediário (repetido 8 vezes) e de saída. Todas as camadas de convolução e convolução separável são seguidas pela *batch normalization*, além disso, todas as convoluções separáveis usam um multiplicador de profundidade 1 (CHOLLET, 2017). Ao todo a Xception possui 36 camadas convolucionais para a extração de atributos, seguidas por uma camada de regressão logística. De forma opcional, é possível inserir camadas totalmente conectadas entre as camadas convolucionais e a camada de regressão logística. As 36 camadas convolucionais estão organizadas através de 14 módulos, que possuem conexões residuais lineares, exceto para o primeiro e o último módulo. As conexões residuais são essenciais para a convergência da rede, tanto em termos de velocidade quanto para o desempenho final de classificação (CHOLLET, 2017).

A Inception V3 e a Xception possuem 23,9 e 22,9 milhões de parâmetros treináveis, respectivamente (ver Tabela 9). Apesar da Xception possuir uma quantidade de parâmetros da mesma ordem da Inception V3, Chollet (2017) constatou que a Xception possui um desempenho um pouco melhor no conjunto de dados ImageNet (DENG et al., 2009) e um desempenho consideravelmente melhor no conjunto de dados interno da Google, JFT,

Figura 38 – Arquitetura da Xception.



Fonte: Adaptado de Chollet (2017).

(HINTON; VINYALS; DEAN, 2015).

2.5 Transferência de aprendizagem

No método tradicional de treinamento de redes neurais artificiais, discutido até então, as redes neurais são treinadas para uma tarefa específica. Isto é, os dados de treinamento e os dados futuros, precisam ter o mesmo espaço de características e a mesma distribuição (PAN; YANG, 2009). Sendo assim, caso haja a necessidade de aplicar a rede em outro contexto ou problema, é necessário treiná-la do zero, utilizando um conjunto de dados específico para essa nova situação.

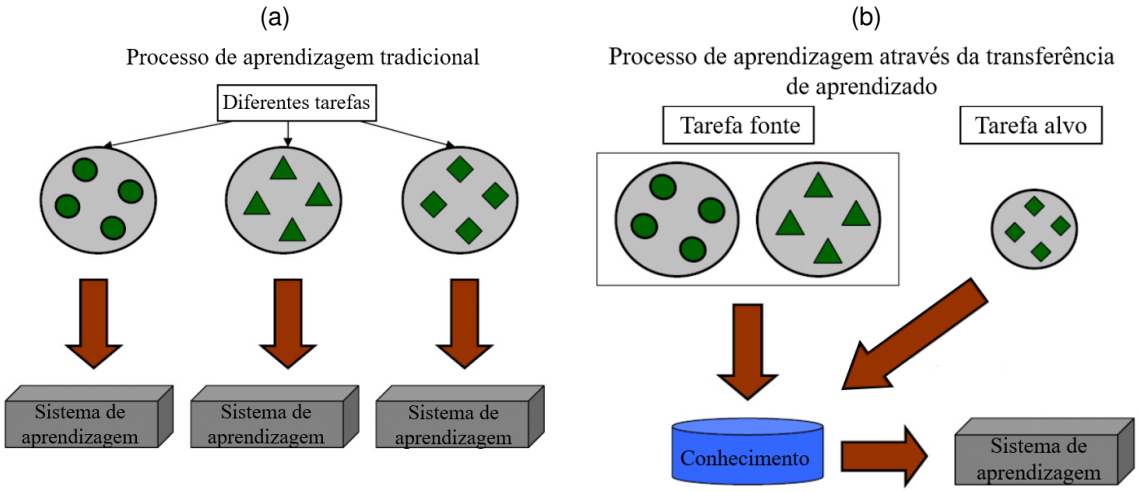
Por outro lado, o processo de aprendizagem dos humanos não acontece dessa maneira. Nós possuímos uma inerente capacidade de transferir conhecimento entre várias áreas. Ao aprendermos uma determina tarefa, somos capazes de utilizar esse conhecimento para realizar outras tarefas similares. Quanto mais similar uma nova tarefa for de tarefas já conhecidas, mais fácil será aprender a nova atividade (SARKAR, 2018). Por exemplo, uma pessoa que sabe tocar violão terá facilidade ao aprender tocar guitarra; saber andar

de bicicleta facilitará o processo de aprendizagem de andar de motocicleta. Diferente do método de aprendizagem tradicional das RNAs, um estudante ao cursar uma disciplina de matemática avançada, não precisará estudar todos os princípios da matemática do zero para compreender os novos conteúdos.

Desta forma, as pessoas são capazes de aplicar um conhecimento aprendido previamente para resolver novos problemas de forma mais rápida ou com soluções melhores. E esta é a motivação da transferência de aprendizagem, a qual permite que o domínio, as tarefas e distribuições utilizadas no processo de treinamento e teste de uma RNA sejam diferentes (PAN; YANG, 2009).

Em outras palavras, através da transferência de aprendizagem, é possível aplicar uma RNA para um problema diferente para o qual ela foi treinada. Isto é, utilizar uma rede pré-treinada em um conjunto de dados para resolver um problema distinto (ROSEBROCK, 2019). A Figura 39a exemplifica o método tradicional de treinamento, onde para diferentes tarefas é necessário treinar redes do zero para cada tarefa. Já a Figura 39b mostra a abordagem através da transferência de aprendizagem, onde é possível utilizar uma rede treinada para resolver um problema novo, seja de uma forma mais rápida, mais eficiente ou que exija um conjunto de treinamento menor.

Figura 39 – (a) Processo de aprendizagem de RNAs tradicional e (b) através da transferência de aprendizagem.



Fonte: Adaptado de Pan e Yang (2009).

No contexto da aprendizagem profunda, a utilização da transferência de aprendizagem pode ser crucial, especialmente para problemas cujo conjunto de dados é pequeno. Redes profundas são complexas e possuem muitos parâmetros a serem ajustados durante

o seu processo de treinamento, o que exige conjuntos de dados com um grande número de exemplos rotulados para um bom treinamento supervisionado. Contudo, a criação de um conjunto de dados pode ser custoso e demandar muito tempo, acima de tudo, quando se tratando de aplicações médicas. Em outras palavras, conseguir um conjunto de dados para um problema médico que seja grande o suficiente para treinar uma rede profunda pode ser inviável. Além do que, as redes neurais convolucionais pode demandar um alto tempo de treinamento. Mesmo utilizando supercomputadores, pois as redes são algoritmos sequenciais. Isto é, uma camada começa a trabalhar apenas quando a camada anterior termina o seu trabalho. Tal processo dificulta a implementação de estratégias de *high performance*.

Para apresentar uma definição formal da transferência de aprendizagem, é necessário primeiro definir o conceito de domínio e tarefa (PAN; YANG, 2009).

Um domínio $\mathcal{D}$ é composto por dois componentes, um espaço de características $\mathcal{X}$ e uma distribuição marginal de probabilidade $P(X)$, onde $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$ (PAN; YANG, 2009). Ou seja, X representa um possível conjunto de treinamento, o qual possui as instâncias $x_1, \ldots, x_n$. Enquanto que $\mathcal{X}$ é o conjunto de todas as possíveis instâncias do problema.

Dado um domínio, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, uma tarefa consiste de dois componentes: um espaço de rótulos (ou classes) $\mathcal{Y}$ e uma função preditiva objetiva f, podendo ser denotada por $\mathcal{T} = \{\mathcal{Y}, f\}$. Em outras palavras, $\mathcal{Y}$ representa o conjunto de todos os rótulos possíveis do problema. A função f é um componente que não se consegue observar, mas a rede pode treinar através dos dados de treinamentos, que consiste dos pares $\{x_i, y_i\}$, onde $x_i \in \mathcal{X}$ e $y_i \in \mathcal{Y}$. A função f pode ser utilizada para predizer o correspondente rótulo $f(x)$, de uma instância x, além disso, do ponto de vista probabilístico, ela pode ser denotada também como $P(y|x)$ (PAN; YANG, 2009).

Segundo Pan e Yang (2009), a formatação mais popular entre os trabalhos da literatura, é a situação quando se tem um domínio fonte (também chamado de *source domain*), $\mathcal{D}_S$, e um domínio alvo (também chamado de *target domain*), $\mathcal{D}_T$. Ou seja, é denotado o conjunto fonte como $\mathcal{D}_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_{n_S}}, y_{S_{n_S}})\}$, onde $x_{S_i} \in \mathcal{X}_S$ é uma instância e $y_{S_i} \in \mathcal{Y}_S$, seu respectivo rótulo. Similarmente, o domínio alvo é dado por $\mathcal{D}_T = \{(x_{T_1}, y_{T_1}), \ldots, (x_{T_{n_T}}, y_{T_{n_T}})\}$, onde a entrada $x_{T_i} \in \mathcal{X}_T$ e $y_{T_i} \in \mathcal{Y}_T$ é a sua correspondente saída. Na maioria dos casos $0 \leq n_T \ll n_S$. Agora então, pode-se definir a

transferência de aprendizagem.

Definição: (*Transferência de Aprendizagem*) Dado um domínio fonte $\mathcal{D}_S$ e uma tarefa de aprendizagem $\mathcal{T}_S$, um domínio alvo $\mathcal{D}_T$ e uma tarefa de aprendizagem $\mathcal{T}_T$, a transferência de aprendizagem busca melhorar a aprendizagem da função preditiva alvo f_T em $\mathcal{D}_T$ usando o conhecimento em $\mathcal{D}_S$ e $\mathcal{T}_S$, onde $\mathcal{D}_S \neq \mathcal{D}_T$, ou $\mathcal{T}_S \neq \mathcal{T}_T$.

Como um domínio é definido pelo par $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a condição $\mathcal{D}_S \neq \mathcal{D}_T$ implica que ou $\mathcal{X}_S \neq \mathcal{X}_T$ ou $P_S(X) \neq P_T(X)$. Similarmente, uma tarefa é o par $\mathcal{T} = \{\mathcal{Y}, P(\mathcal{Y}|\mathcal{X})\}$, onde a condição $\mathcal{T}_S \neq \mathcal{T}_T$ implica que ou $\mathcal{Y}_S \neq \mathcal{Y}_T$ ou $P(\mathcal{Y}_S|\mathcal{X}_S) \neq P(\mathcal{Y}_T|\mathcal{X}_T)$. Quando os domínios fonte e alvo são os mesmos, ou seja, $\mathcal{D}_S = \mathcal{D}_T$, e as tarefas de aprendizagem são as mesmas ($\mathcal{T}_S = \mathcal{T}_T$), o problema de aprendizagem se torna em um problema de aprendizagem de máquina tradicional (PAN; YANG, 2009).

Basicamente, no contexto da aprendizagem profunda é possível aplicar a transferência de aprendizagem de duas maneiras: a transferência através da extração de atributos e a transferência via ajuste fino (ROSEBROCK, 2019).

Na modalidade de extração de atributos, a rede pré-treinada é utilizada como um extrator de atributos arbitrário. Assim, a instância de entrada é propagada pela rede até uma camada preestabelecida. A saída desta camada é obtida como atributos que poderão ser usados para treinar um classificador qualquer (ROSEBROCK, 2019).

Já na modalidade de ajuste fino, apenas as camadas totalmente conectadas do modelo são treinadas do zero com o conjunto de dados alvo. As outras camadas são mantidas fixas ('congeladas') com os parâmetros pré-treinados (ROSEBROCK, 2019).

Para as CNNs, um conjunto de dados bastante utilizado para pré-treinar as redes é o ImageNet (DENG et al., 2009). Um dos maiores conjunto de imagens para reconhecimento visual, ImageNet possui mais de 1,2 milhões de imagens 256$\times$256 categorizadas em 1000 objetos. Onde os objetos podem aparecer nas imagens de forma obscura, pequenos, parcialmente, e fora de contexto (ambiente desordenado) (DENG et al., 2009; SHIN et al., 2016). A Figura 40 mostra alguns exemplos de imagens da classe 'bola de tênis' do ImageNet.

Neste trabalho, foi utilizado a modalidade de extração de atributos da transferência de aprendizagem. Para isto, foram utilizadas CNNs pré-treinadas no conjunto de imagens ImageNet. Os atributos foram obtidos na última camada das CNNs logo antes das camadas

Figura 40 – Exemplos de imagens pertencentes à classe ’bola de ténis’ do ImageNet.

Fonte: Shin et al. (2016).

completamente conectadas.

2.6 Seleção de atributos

Uma das mais importantes tarefas de reconhecimento de padrões e classificação é a redução da dimensionalidade das instâncias (CHEN; CHEN; CHEN, 2013). A seleção de atributos é o processo responsável por identificar e remover o máximo possível informações irrelevantes e redundantes do conjunto de dados (HALL, 1999). Isto é feito ao procurar por todas as possíveis combinações de atributos qual o subconjunto é o melhor para predição e classificação (BOUCKAERT et al., 2016). O objetivo da seleção de atributos é escolher um subconjunto do conjunto de atributos que elimina características desnecessárias que possuem pouca ou nenhuma informação preditiva ou que são fortemente correlatas (CHEN; CHEN; CHEN, 2013).

Teoricamente, quanto mais atributos maior será o poder descriminador do classificador, contudo, a experiência prática com algoritmos de aprendizagem vem mostrando que isso não é o que acontece (HALL, 1999). A representação das instâncias em um espaço de alta dimensão pode resultar em dados esparsos no mapa das características. Em situações extremas os dados podem ficar tão distantes entre si em um mapeamento esparso que instâncias da mesma classe se tornam pouco similares. Desta maneira, para um determinado número de instâncias fixo em uma base de dados, o aumento do número de atributos acarreta no aumento do erro do classificador. Esse fenômeno recebe o nome de maldição da dimensionalidade (LIMA, 2020). Porém, o termo maldição da dimensionalidade se refere a todos os fenômenos que acontecem com dados de alta dimensão que interfiram negativamente no desempenho dos algoritmos de aprendizado (VERLEYSEN; FRANÇOIS,

2005).

A redução da dimensionalidade dos dados pode permitir que o algoritmo de aprendizagem possa operar mais rápido e mais eficiente, diminuindo o tempo computacional, e em alguns casos, aumentando a acurácia (HALL, 1999; CHEN; CHEN; CHEN, 2013). Além disso, um conjunto com baixa dimensionalidade necessitará de um modelo de RNA mais simples. Sendo assim, a seleção de atributos é indicada para problemas que possuem um grande número de atributos, ao mesmo tempo em que é desconhecida a relevância de cada atributo (CASTRO et al., 2004). Ou seja, a seleção de atributos ajuda no entendimento dos dados além de reduzir o custo computacional, o efeito da maldição da dimensionalidade e melhorar o desempenho do preditor (CHANDRASHEKAR; SAHIN, 2014).

Para um conjunto de dados com um grande número de atributos é possível que exista atributos correlatos entre si. No caso onde dois atributos são fortemente correlatos, apenas um atributos é suficiente para descrever os dados. Variáveis dependentes não fornecem nenhuma informação extra as classes, servindo apenas como ruídos para o preditor, sendo adequado sua remoção para melhorar o desempenho de classificação. Assim, a informação total pode ser obtida a partir de um número menor de atributos únicos que contém máxima informação discriminatória sobre as classes (CHANDRASHEKAR; SAHIN, 2014). Em algumas aplicações, variáveis que possuem nenhuma correlação com as classes servem apenas como ruído, podendo introduzir um viés ao preditor e reduzir o desempenho de classificação, isto acontece quando possui uma falta de informação sobre o processo estudado. Neste caso, a seleção de atributos pode fornecer um discernimento sobre o processo ao apontar tais atributos sem relevância (CHANDRASHEKAR; SAHIN, 2014).

Basicamente, a seleção de atributos pode ser realizada através dos métodos *Filter* (filtro), *Wrapper* (invólucro) e *Embedded* (embutido) (CHANDRASHEKAR; SAHIN, 2014; JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015). Os métodos *filters* usam técnicas de ranqueamento das variáveis como critério para a seleção de atributos por ordenamento. Essa abordagem é simples e bons resultados são obtidos em aplicações práticas (CHANDRASHEKAR; SAHIN, 2014). Um critério de ranqueamento adequado é utilizado para pontuar aos atributos e um limite é utilizado para remover as variáveis abaixo deste limite (CHANDRASHEKAR; SAHIN, 2014; JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015). A grosso modo, as medidas de filtragem de atributos são baseadas em: informação, distância, consistência, similaridade, e medidas

estatísticas (JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015).

Métodos *wrappers* utilizam o desempenho do preditor como função objetivo para avaliar o subconjunto de atributos, o preditor é então utilizado como um avaliador caixa preta. A avaliação é repetida para cada subconjunto, e a geração de subconjuntos são dependentes da estratégia de busca, da mesma forma que os métodos *filters* (CHANDRASHEKAR; SAHIN, 2014; JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015). Como os métodos *wrappers* utilizam um modelo real como método de avaliação, foi provado empiricamente que métodos *wrappers* obtêm subconjuntos com melhores resultados do que os obtidos pelos *filters* (JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015). Por outro lado, por não ser moldado especificamente para um preditor, um subconjunto encontrado por métodos *filter* não é ótimo para um determinado classificador, contudo o subconjunto não é dependente de nenhum classificador (HALL, 1999). Métodos *wrappers* são bem mais lentos que os *filters*, justamente por depender de demandas de recursos do algoritmo de preditor (JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015). Para cada avaliação de um subconjunto é gerado um novo modelo o qual será treinado e testado para obter a acurácia do classificador. Desta forma, boa parte da execução do algoritmo é gasta em treinar o preditor (CHANDRASHEKAR; SAHIN, 2014).

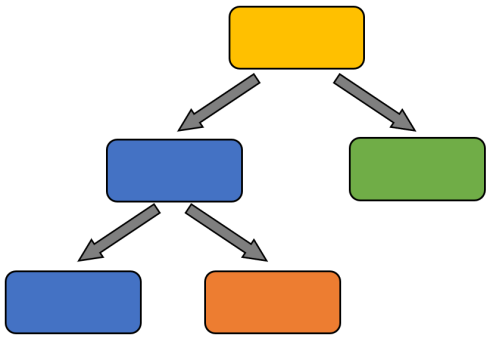
Métodos *embedded* buscam reduzir o tempo computacional utilizado ao treinar classificadores para diferentes subconjuntos como é feito pelos métodos *wrappers* (CHANDRASHEKAR; SAHIN, 2014). Isso é feito ao desempenhar a seleção de atributos durante a execução do algoritmo de modelagem. Métodos *embedded* são, portanto, incorporados no algoritmo como sua funcionalidade normal ou estendida (JOVIĆ; BRKIĆ; BOGUNOVIĆ, 2015).

A *Random Forest* (RF) é um algoritmo de classificação e regressão que pode ser utilizado também para medir a importância de variáveis e para seleção de atributos (QI, 2012; HASAN et al., 2016). Na RF a seleção de atributos ocorre durante a construção das regras de classificação (QI, 2012). Portanto, a *Random Forest* é um método *embedded* (CHANDRASHEKAR; SAHIN, 2014). Segundo Sandri e Zuccolotto (2006) a seleção de atributos baseada em *Random Forest* identifica um número menor de atributos relevantes e permite a construção de um modelo mais parcimonioso, mas mantendo o desempenho preditivo. Na seção a seguir é descrito como as *Random Forests* funcionam e como elas podem ser utilizadas na seleção de atributos.

2.7 Random Forest

Random Forest (RF) é um algoritmo supervisionado da aprendizagem de máquina baseado em árvores de decisão (BREIMAN, 2001; QI, 2012). A arquitetura de uma árvore de decisão é baseada em nós, os quais são organizados de uma maneira hierárquica, como mostrado na Figura 41. O ponto de partida de uma árvore de decisão é o nó raiz (nó superior amarelo da Figura 41), que possui a maior posição hierárquica. A partir do nó raiz partem dois ramos onde se encontram mais nós. Esses nós, por sua vez, podem se dividir, até atingir o final da árvore com os nós folhas (ou nós terminais), que não possuem ramificações. Os nós folhas representam decisões, e por isso, fornecem a saída da árvore de decisão. Dessa maneira, uma árvore de decisão toma decisões após seguir um caminho que começa no nó raiz e vai até um nó folha (GOMES et al., 2020).

Figura 41 – Esquema de arquitetura de uma árvore de decisão.

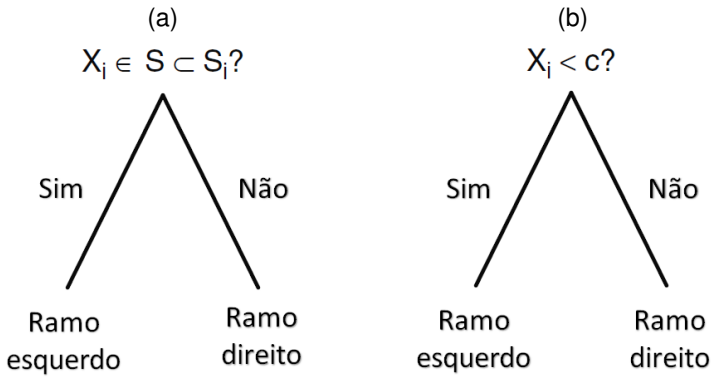


Fonte: O Autor (2022).

A ramificação de um nó é feita através de uma pergunta com respostas ”sim” ou ”não”, que separa um atributo de acordo com um dado valor. Na Figura 42 são mostrados ramos para a separação de variáveis categóricas e contínuas. Para variáveis categóricas, Figura 42a, se S é o conjunto de todas as possíveis categorias para uma variável categórica, a pergunta considerada é se atributo a ser avaliado, X_i, está contido em um subconjunto S_i de S. Se verdadeiro, então a divisão vai para o ramo esquerdo, caso contrário vai para o da direita (CUTLER; CUTLER; STEVENS, 2012). Para variáveis contínuas, Figura 42b, é avaliado se o atributo, X_i, é menor do que um dado valor c. Em caso afirmativo, é tomado o ramo esquerdo, senão é tomado o ramo direito.

A *Random Forest* é um modelo composto por um conjunto de árvores de decisão, onde cada árvore depende de uma coleção de variáveis aleatórias (CUTLER; CUTLER; STEVENS, 2012). Cada árvore de decisão que compõe a *Random Forest* é treinada com

Figura 42 – Separação de variáveis, em a) para variáveis categóricas e b) para variáveis contínuas.



Fonte: Adaptado de Cutler, Cutler e Stevens (2012).

um subconjunto do conjunto de dados de treinamento. Esse subconjunto recebe o nome de conjunto *bootstrap dataset*. O *bootstrap dataset* é um conjunto de mesmo tamanho do conjunto original, construído de forma aleatória através de um sorteio com reposição, assim, permitindo a possibilidade de haver instâncias repetidas. Além disso, o treinamento de cada árvore é feito apenas com uma parcela de atributos aleatoriamente selecionados do *bootstrap dataset*. Sendo assim, na RF a aleatoriedade do modelo é realizada de duas maneiras (CUTLER; CUTLER; STEVENS, 2012), pela concepção do *bootstrap dataset* e pelos atributos selecionados. O objetivo disso, é obter um conjunto de classificadores ou regressores distintos entre si, o que aumenta a robustez do método.

Uma vez que o conjunto de árvores de decisão da RF é treinado, essas árvores formam um comitê de classificadores ou regressores para o dado problema. Sendo assim, na avaliação de uma nova instância, o dado de entrada será avaliado por todas as árvores de decisão da RF. Para um problema de classificação, a saída da RF será tomada pela classe mais votada entre todas as árvores. Enquanto que para um problema de regressão, a saída da RF é tomada pela média aritmética das saídas de todas as árvores (CUTLER; CUTLER; STEVENS, 2012).

As instâncias do conjunto de dados que não fazem parte do *bootstrap dataset* formam o *out-of-bag dataset*. Essas instâncias não participam do processo de treinamento das árvores, e podem ser utilizadas tanto para estimar o erro de generalização do método, como também estimar a importância das variáveis do problema (CUTLER; CUTLER; STEVENS, 2012). Como cada árvore da RF possui um *bootstrap dataset* específico, então cada árvore tem um *out-of-bag dataset* diferente. A estimativa do erro da RF é feita ao avaliar uma

instância do conjunto *out-of-bag* a todas as árvores que não a utilizaram durante o seu processo de treinamento. Ou seja, nesse processo são utilizadas apenas aquelas árvores que essa amostra faz parte dos seus conjuntos *out-of-bag*. Para essa amostra, a saída da RF considerada, será a classe mais votada. A proporção de amostras *out-of-bag* classificadas incorretamente é chamada de erro *out-of-bag* (CUTLER; CUTLER; STEVENS, 2012).

A medição da importância dos atributos pela *Random Forest* é feita a partir das amostras *out-of-bag* e as predições da RF para essas amostras. Assim, para medir a importância de um atributo k, os valores desse atributos são permutados aleatoriamente entre as amostras *out-of-bag*. Em seguida, a RF avalia essas amostras com os novos valores de k. Então, a novas predições são comparadas com as predições obtidas para o dado original (CUTLER; CUTLER; STEVENS, 2012). Nesse processo, o quão maior for o impacto das permutações nas predições da RF, maior será a importância do atributo k. Em outras palavras, a diferença entre os erros das predições para os dados originais e os dados permutados dará a medida de importância dos atributos (CUTLER; CUTLER; STEVENS, 2012). Em função dessa medida, é possível selecionar os atributos que são mais relevantes, ou de maior poder preditivo para o problema. O Algoritmo 5 descreve o processo para a medição da importância de variáveis.

Algoritmo 5 Medição da importância de atributos

1: **para** cada atributo k **fazer**
2: **para** cada instância i do conjunto de dados **fazer**
3: Avaliar a predição das árvores da RF a quais possuem a instância i no seu conjunto *out-of-bag*;
4: **fim para**
5: **para** para cada árvore j da RF **fazer**
6: Permutar aleatoriamente os valores da variáveis k entre as instâncias que compõem conjunto *out-of-bag* da árvore j;
7: Avaliar a predição das árvores da RF para todas as instâncias do seu conjunto *out-of-bag*;
8: **fim para**
9: **para** cada instância i do conjunto de dados **fazer**
10: Comparar as predições com as amostras reais do conjunto *out-of-bag* com as predições das amostras permutadas.
11: **fim para**
12: **fim para**

Fonte: Adaptado de Cutler, Cutler e Stevens (2012).

2.8 *Synthetic Minority Oversampling TEchnique (SMOTE)*

Uma base de dados é chamada desbalanceada quando o número de instâncias entre as classes não são aproximadamente o mesmo (CHAWLA et al., 2002). Isto é, para bases desbalanceadas uma ou mais classes possuem muito mais instâncias do que a(s) outra(s). As classes com maior número de instâncias são chamadas de classes majoritárias, já as outras são as classes minoritárias.

O uso de classificadores em bases desbalanceadas pode enviesar a classificação a favor da classe majoritária. O viés pode ser ainda maior para dados de alta dimensão (BLAGUS; LUSA, 2013). Resultados, mal interpretados, de classificação em base desbalanceadas podem causar uma falsa sensação de bons resultados. Pois a saída do classificador pode ser na maioria das vezes a favor da classe majoritária. Como a classe majoritária possui mais instâncias, a taxa de acerto será alta.

Os fenômenos da natureza não acontecem de forma balanceada. Por exemplo, para cada paciente que possui câncer de mama, existem um número consideravelmente maior de paciente sadias. Assim, utilizar uma base contendo uma distribuição desigual de instâncias entre as classes para treinar um classificador fará com que o algoritmo tenda para a classe majoritária. Assim, classificadores treinados em bases desbalanceadas tem baixo poder de generalização (BLAGUS; LUSA, 2013).

Uma maneira de lidar com bases desbalanceadas é promover uma sobreamostragem da classe minoritária. Isto é, balancear a base através da inserção de novas instâncias da classe minoritária. O *Synthetic Minority Oversampling TEchnique* (SMOTE) é um método de geração de instâncias sintéticas para diminuir o efeito do desequilíbrio de classes em bases desbalanceadas (BLAGUS; LUSA, 2013). O SMOTE foi primeiramente proposto por CHAWLA et al. em 2002, o qual usa a combinação linear de duas instâncias da classe minoritária para construir uma nova instância sintética (CHAWLA et al., 2002; BLAGUS; LUSA, 2013). Através do SMOTE, considerando $\boldsymbol{x}$ uma instância da classe minoritária, uma nova instância sintética $\boldsymbol{s}$ é obtida através da seguinte equação

$$\boldsymbol{s} = \boldsymbol{x} + u \cdot (\boldsymbol{x}_R - \boldsymbol{x}), \tag{2.69}$$

onde u é um número aleatório gerado dentro do intervalo [0, 1]; e $\boldsymbol{x}_R$ é escolhido aleatori-

amente entre as k instâncias da classe minoritária vizinhas de $\boldsymbol{x}$. Geralmente, é utilizado k igual à 5 (CHAWLA et al., 2002; BLAGUS; LUSA, 2013). Desta maneira, a instância $\boldsymbol{s}$, dada pela Equação 2.69, se encontrará em algum ponto aleatório ao longo do segmento de linha entre $\boldsymbol{x}$ e $\boldsymbol{x}_R$, isso faz com que a região de decisão da classe minoritária se torne mais geral (CHAWLA et al., 2002).

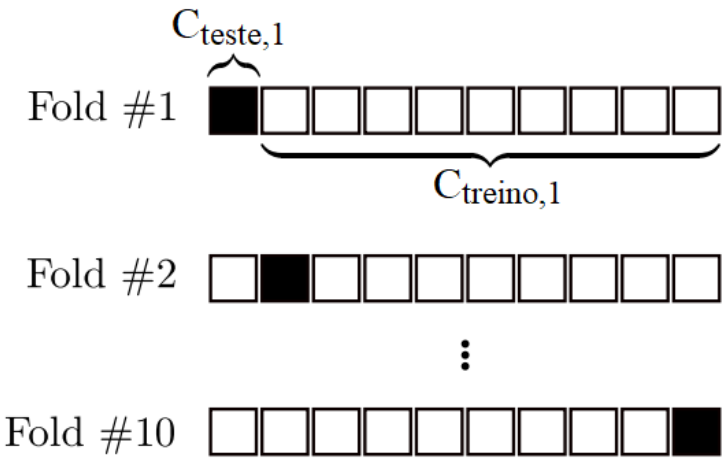
2.9 Validação cruzada

O treinamento e validação de uma rede neural artificial pode acontecer de diferentes modos para um dado conjunto de dados. Uma das abordagens mais simples é a separação do conjunto em um conjunto de treinamento e outro para testes, onde normalmente usa-se 60% dos dados para treino e os outros 40% para teste.

Um método bastante utilizado neste contexto é a validação cruzada, sobretudo quando o conjunto de dados é pequeno (HAYKIN, 2001). A validação cruzada é um dos métodos mais utilizados para avaliar a capacidade de generalização de um modelo (BERRAR, 2019). Uma das abordagens da validação cruzada é o método K-*fold*. Nessa abordagem o conjunto de dados é dividido em $K > 1$ subconjuntos. Em seguida a rede é treinada com a união de $K - 1$ desses subconjuntos e testada com o subconjunto remanescente. Este procedimento é então repetido K vezes de modo que cada subconjunto seja utilizado uma vez como conjunto de teste. Assim, todos os dados do conjunto é utilizado ora para treinamento do modelo, ora para teste. O desempenho da rede é dada pelo erro quadrático médio dos K testes (HAYKIN, 2001). Desta maneira, a acurácia final do modelo não será interferida pelos conjuntos reservados para treino e teste. A Figura 43 apresenta um esquemático da validação cruzada com 10-*fold*, onde $C_{treino,1}$ representa o conjunto de treinamento formado pela união de 9 subconjuntos do conjuntos de dados, e $C_{teste,1}$ é o subconjunto remanescente, a ser utilizado para o treinamento. Esse processo é repetido 10 vezes de modo que cada subconjunto seja utilizado como conjunto de teste.

A validação cruzada permite uma estimativa de desempenho precisa para novos dados (REFAEILZADEH; TANG; LIU, 2016; BERRAR, 2019), dessa maneira, tal método é bastante utilizado durante a fase de construção e avaliação de modelos preditivos. Porém na construção de um modelo final para ser utilizado com casos reais futuros, o modelo é treinado com todo o conjunto de dados. Nesse último caso, não é possível a validação

Figura 43 – Validação cruzada 10-*fold*.



Fonte: Adapatado de Berrar (2019).

do modelo de forma cruzada (BERRAR, 2019). Contudo isso não é mais necessário, pois o modelo já foi testado previamente para o problema durante sua concepção. Na aprendizagem de máquina, o valor de K igual a 10 é mais comum (REFAEILZADEH; TANG; LIU, 2016; BERRAR, 2019). Como necessita treinar a RNA K vezes, a validação cruzada demanda um grande consumo computacional, sendo esta uma desvantagem do método.

Uma outra abordagem da validação cruzada é o método deixe um fora (ou do inglês *leave-one-out cross-validation*) que é indicado para conjunto de dados extramente pequenos. Considerando N a quantidade de exemplos do conjunto, no método deixe um fora utiliza-se $N - 1$ pontos para formar o conjunto de treinamento e a validação é feita sobre o ponto remanescente. Da mesma maneira do que a abordagem convencional, o procedimento será repetido N vezes e o erro quadrático médio é utilizado como métrica de avaliação do desempenho da RNA (HAYKIN, 2001).

2.10 Métricas de avaliação

Nesta seção são discutidas as métricas que foram utilizadas para avaliar o desempenho dos métodos de classificação aplicados em imagens de termografia. Antes de definir as métricas de avaliação é necessário definir os conceitos de verdadeiro positivo, falso positivo, verdadeiro negativo e falso negativo. Para isto, considere um problema de classificação binária que possui as classes $\mathcal{C}_1$ e $\mathcal{C}_2$, como por exemplo em um contexto onde se quer classificar paciente com uma determinada doença ($\mathcal{C}_1$) ou sadios ($\mathcal{C}_2$). Nesta situação se

tem que:

- Verdadeiro positivo (VP): Quando o classificador identifica um ponto como pertencente à classe $\mathcal{C}_1$ e o ponto de fato pertence a essa classe (SKANSI, 2018). No contexto de diagnóstico médico isso significa quando um indivíduo com uma doença é corretamente diagnosticado.
- Falso positivo (FP): Quando o classificador considera um ponto como pertencente à classe $\mathcal{C}_1$ mas na verdade pertence à $\mathcal{C}_2$ (SKANSI, 2018). Isto é equivalente quando um indivíduo sadio é diagnosticado como doente.
- Verdadeiro negativo (VN): Quando um ponto da classe $\mathcal{C}_2$ é classificado como sendo da classe $\mathcal{C}_2$ (SKANSI, 2018). Por exemplo quando um indivíduo sadio é diagnosticado como sadio.
- Falso negativo (FN): Quando um ponto da classe $\mathcal{C}_2$ é classificado como sendo da classe $\mathcal{C}_1$ (SKANSI, 2018). Por exemplo, indivíduo com a doença mas diagnosticado como sadio.

Apesar desses termos terem sido definidos para uma classificação binária, seus significados podem ser estendidos para um contexto de classificações multiclasse. A seguir os termos VP, VN, FP e FN representam os números de verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo, respectivamente.

Sensibilidade

A capacidade de um teste identificar uma doença em pacientes que a possuem é chamada de sensibilidade. Em outras palavras, sensibilidade é a probabilidade de um paciente ser corretamente diagnosticado quando estiver doente (MOGHBEL; MASHOHOR, 2013). A sensibilidade é calculada pela equação:

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \tag{2.70}$$

Especificidade

Especificidade é a probabilidade de um teste em determinar que uma pessoa está sem a doença quando ela realmente está sadia (MOGHBEL; MASHOHOR, 2013). Seu calculo é feito da seguinte forma:

$$\text{Especificidade} = \frac{VN}{VN + FP} \tag{2.71}$$

Acurácia

A acurácia é uma métrica que combina os efeitos da sensibilidade e especificidade. Seu objetivo é avaliar a capacidade de um teste de diagnosticar como doente, as pessoas que realmente possuem a doença. E de diagnosticar como sadias, as pessoas que estão livres da doença de fato (MOGHBEL; MASHOHOR, 2013). A acurácia é dada pela equação:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.72}$$

Precisão

A precisão é a fração de predições positivas que realmente são positivas, dadas pela equação:

$$\text{Precisão} = \frac{VP}{VP + FP}. \tag{2.73}$$

Estatística kappa

Antes de definir a estatística kappa é necessário definir a matriz de confusão. Para um conjunto de classes de interesse $\Omega = \{\mathcal{C}_1,\ \mathcal{C}_2,\ \ldots, \mathcal{C}_m\}$ define-se a matriz de confusão como a matriz $\boldsymbol{T} = [t_{i,j}]_{m \times m}$, onde cada elemento $t_{i,j}$ representa o número de objetos pertencentes a classe $\mathcal{C}_j$ mas que foram classificados como $\mathcal{C}_i$ (AZEVEDO et al., 2015). Para um problema de classificação binária, a matriz de confusão é uma matriz 2×2 como a mostrada na Tabela 7 (SKANSI, 2018).

Tabela 7 – Exemplo de uma matriz de confusão para um problema de classificação binária.

	Classificado como $\mathcal{C}_1$	Classificado como $\mathcal{C}_2$
Pertence à $\mathcal{C}_1$	VP	FN
Pertence à $\mathcal{C}_2$	FP	VN

Fonte: Adaptado de Skansi (2018).

É importante salientar que uma vez que se tem a matriz de confusão é possível calcular acurácia, sensibilidade e especificidade (SKANSI, 2018). Como a acurácia representa a relação entre o número de objetos corretamente classificados com os incorretamente classificados e sabendo que os elementos da diagonal principal da matriz de confusão ($t_{i,i}$) representam o número de objetos corretamente classificados, enquanto que os outros elementos ($t_{i,i}$, para $i \neq j$) representam os incorretamente classificados, pode-se redefinir a acurácia da seguinte maneira:

$$\rho_v = \frac{\sum_{i=1}^{m} t_{i,i}}{\sum_{i=1}^{m} \sum_{j=1}^{m} t_{i,j}} \tag{2.74}$$

A estatística kappa κ é então dada pela equação:

$$\kappa = \frac{\rho_v - \rho_z}{1 - \rho_z} \tag{2.75}$$

onde:

$$\rho_z = \frac{\sum_{i=1}^{m} (\sum_{j=1}^{m} t_{i,j})(\sum_{j=1}^{m} t_{j,i})}{(\sum_{i=1}^{m} \sum_{j=1}^{m} t_{i,j})^2} \tag{2.76}$$

A estatística kappa assume valores dentro do intervalo [0, 1]. Na Tabela 8 são dados subintervalos de [0, 1] e suas respectivas classificações de desempenho do processo de classificação.

Tabela 8 – Intervalos de valores para o indíce kappa e seus desempenhos correspondentes

kappa	Desempenho
$\kappa < 0,0$	Muito Ruim
$0,0 < \kappa \leq 0,2$	Ruim
$0,2 < \kappa \leq 0,4$	Razoável
$0,4 < \kappa \leq 0,6$	Bom
$0,6 < \kappa \leq 0,8$	Muito Bom
$0,8 < \kappa \leq 1,0$	Excelente

Fonte: Adaptado de Azevedo et al. (2015).

3 TRABALHOS RELACIONADOS

Este trabalho se enquadra numa linha de pesquisa desempenhada por pesquisadores dos departamentos de Engenharia Mecânica e Biomédica da UFPE, onde uma série de projetos de pós-graduação foram desenvolvidos. Desta maneira, este capítulo é dedicado tanto a revisão de trabalhos relacionados a esta tese publicados nos últimos anos, como também a revisão de alguns dos trabalhos desenvolvidos na UFPE.

3.1 Estado da arte

Muitos pesquisadores vêm investigando o uso de imagens de termografia associadas com avaliação automática como método complementar no rastreamento do câncer de mama. Neumann et al. (2016) propuseram uma metodologia de pré-processamento para balancear conjunto de dados compostos por imagens termográficas obtidas por um dispositivo móvel. A metodologia é baseada em *standardization*, *Synthetic Minority Oversampling Technique* (SMOTE) (CHAWLA et al., 2002; BLAGUS; LUSA, 2013) e subamostragem. Eles utilizaram atributos de contorno para treinar *Decision Tree Classifier* (DTC), *Naïve Bayes*, *Random Forest*, e Máquina de Vetor de Suporte (SVM). As imagens foram rotuladas como patológicas e não patológicas. Neumann et al. (2016) concluíram que a *standardization* melhorou os resultados de DTC e SVM. O SMOTE e a subamostragem melhoraram a sensibilidade da DTC, *Random Forest* e SVM, ao mesmo tempo que diminuiu ligeiramente suas especificidades. Por outro lado, *Naïve Bayes* não teve seu desempenho melhorado pelo SMOTE e a subamostragem. Oleszkiewicz et al. (2016), utilizando o mesmo conjunto de dados e o mesmo processo de pré-processamento utilizado por Neumann et al. (2016), treinaram uma SVM para a detecção precoce do câncer de mama. Cada elemento do conjunto de treinamento foram rotulados como patológico ou normal. Eles concluíram que a SVM entregou melhor sensibilidade e especificidade comparado com *Decision Trees*, *Random Forest* e *Naïve Bayes*.

Karim, Mohamed e Ryad (2018) também treinaram um modelo de SVM considerando 50 imagens de termografia de pacientes saudáveis e 30 com algumas alterações. O seu pré-processamento foi baseado em segmentação, análise de textura e matemática morfológica.

Com 93,3% de sensibilidade, 90% de especificidade e 91,25% de acurácia eles concluíram que as imagens de termografia de mama podem ser implementadas como um método complementar como técnica de rastreamento em seu país, Argélia.

Morales-Cervantes et al. (2018) propuseram uma pontuação térmica para indicar anomalias térmicas. Essa pontuação é baseada em assimetrias térmicas e comparações das áreas mais vascularizadas de cada mama. Analisando 206 termogramas de pacientes com suspeita de câncer de mama eles atingiram 100% de sensibilidade e 68,68% de especificidade.

Araújo, Lima e Souza (2014) utilizaram dados intervalares na estrutura de análise de dados simbólicos (*symbolic data analysis* - SDA) para modelar anormalidades de mama a fim de detectar o câncer de mama. Dessa maneira, eles utilizaram variáveis intervalares obtidas pelos valores de temperatura mínimo e máximos extraídos das matrizes morfológicas e térmicas. Eles utilizaram também operadores baseados em dissimilaridades e o critério de Fisher para obter atributos para o processo de classificação. O conjunto de imagens utilizado por eles é uma versão preliminar do conjunto utilizado neste presente trabalho (ver as Seções 2.1 e 4.2.1). Todas as imagens foram adquiridas pelo Hospital das Clínicas da Universidade Federal de Pernambuco (HC-UFPE). Em 2014, o conjunto era composto por imagens de 50 pacientes com massa suspeita, as quais o diagnóstico foram confirmados através exames clínicos. Eles aplicaram a abordagem de extração de atributos proposta para a classificação de imagens de termografia de mama nas seguintes classes: maligna, benigna e cisto. Diferentes classificadores foram considerados para detectar o câncer de mama, alcançando 16% de taxa de classificação incorreta, 85,7% de sensibilidade e 86,5% de especificidade para a classe de maligna.

Em 2018, Santana et al. (2018) trabalharam com as imagens do HC-UFPE para investigar o desempenho de diferentes classificadores utilização descritores de Haralick e Zernike para extração de atributos. Esses extratores são baseados em atributos de textura e geometria, respectivamente. Naquela época, o conjunto de dados continha imagens de 100 pacientes do sexo feminino, sendo 219 imagens de paciente com cisto, 371 com lesões benignas e 235 contendo lesões malignas. Santana et al. (2018) consideraram as imagens obtidas através das posições frontal e lateral, e classificadas como maligna, benigna e cisto. Eles consideraram os seguintes classificadores: *Bayes Net, Naïves Bayes*, *decision tree*, SVM, Perceptron de Multicamadas (MLP), *Random Forest*, *Random Tree* e Máquina de

Aprendizagem Extremo (ELM). Os melhores resultados foram alcançados pela ELM com 71,22% de precisão e 0,6676 de kappa, e MLP com 76,01% e 0,6402 de precisão e kappa, respectivamente.

Dando continuidade ao trabalho realizado por Santana et al. (2018), Rodrigues et al. (2019) aplicaram seleção de atributos no conjunto de dados obtido pelos descritores de Haralick e Zernike. Eles investigaram o desempenho de Algoritmos Genéticos (GA) e Otimização de Enxame de Partículas (PSO) na seleção de atributos para a identificação de lesões mamárias. Eles realizaram uma classificação multiclasse para as classes normal, maligno, benigno e cisto. Originalmente, o conjunto de dados possuía 169 atributos obtidos pelos momentos de Haralick e Zernike. Resultados experimentais apontaram que foi possível alcançar 91,12% de acurácia do problema usando SVM com *kernel* polinomial de grau quatro. Porém, com a aplicação do GA na seleção de atributos foi possível reduzir o número de atributos para 57 e atingir 87,08% de precisão. Por outro lado, usando o PSO, eles encontraram um subconjunto com 60 atributos onde foi possível obter uma precisão de 86,16%. Ambos os resultados usando SVM com *kernel* polinomial de grau cinco.

Como Aprendizagem Profunda é um campo emergente na Inteligência Artificial, vários pesquisadores estão trabalhando no uso de Redes Neurais Convolucionais (CNN) para identificar lesões mamárias em imagens termográficas. Baffa e Lattari (2018) foram um dos pioneiros a usar uma rede neural convolucional para analisar imagens termográficas de mama. O conjunto de imagens utilizado foi o *Database for Mastology Research with Infrared Image* - (DMR) (SILVA et al., 2014) um conjunto de dados amplamente utilizado em vários trabalhos (ROSLIDAR et al., 2019; CHAVES et al., 2020; EKICI; JAWZAL, 2020; MISHRA et al., 2020; MAMBOU et al., 2018; GOGOI et al., 2018; SILVA et al., 2016; SILVA et al., 2020; TELLO-MIJARES; WOO; FLORES, 2019; SÁNCHEZ-RUIZ; OLMOS-PINEDA; OLVERA-LÓPEZ, 2020).

O conjunto DMR possui imagens termográficas obtidas através dos protocolos estático e dinâmico. No protocolo estático é obtido apenas uma imagem por paciente, após a sua temperatura corporal ter entrado em equilíbrio térmico com o meio ambiente controlado. No protocolo dinâmico a pele da paciente passa por um estresse térmico causado por um ventilador elétrico. Em seguida, durante um dado período de tempo, são capturadas ao todo 20 imagens por paciente (BAFFA; LATTARI, 2018). O DMR é um conjunto público que contém imagens termográficas de 287 pacientes sadias e não sadias. As imagens foram

obtidas por uma câmera termográfica FLIR SC-630, e possuem resolução de 640×480 *pixels*. Entre as imagens estáticas, o DMR possui 177 imagens de pacientes sadias e 42 imagens de pacientes com câncer. Do protocolo dinâmico o DMR possui imagens de 95 pacientes sadias e 42 não sadias, o que totaliza 1900 e 840 imagens sadias e não sadias, respectivamente (BAFFA; LATTARI, 2018).

Como o protocolo dinâmico captura 20 imagens por pacientes, Baffa e Lattari (2018) propuseram quatro estratégias para que as imagens sejam consideradas por um modelo convolucional, são elas: 1. Todas as imagens são consideradas em um único dado de entrada, isto é, no mesmo *array*, como se fossem uma mesma imagem; 2. A imagem resultante da média das 20 imagens é calculada, a imagem final é a imagem a ser utilizada para treinar a rede; 3. Para cada paciente, é calculada a imagem resultante da média da primeira e a última imagem. Assim, apenas as imagens com diferença significativa são consideradas. Só a imagem final é considerada para treinar a rede; 4. Calcula-se a imagem resultante da diferença entre a última imagem e a primeira. Apenas essa imagem é utilizada para o treinamento.

Para a classificação das imagens de termografia Baffa e Lattari (2018) propuseram uma arquitetura de CNN. A rede possui duas camadas convolucionais com filtros 5×5 e 32 saídas, seguidas por duas camadas *max-pooling* com tamanho 5×5 e limiar de 3. A camada de saída é uma camada totalmente conectada que classifica os dados nas classes sadia e não sadia. Todas as imagens de entrada são redimensionadas para a resolução 56×56, e o treinamento é feito utilizando validação cruzada. Como os conjuntos de dados utilizados são desbalanceados, possuindo mais dados de pacientes sadias do que não sadias, Baffa e Lattari (2018) executaram o balanceamento dos conjuntos para ambos os protocolos. Imagens artificiais para a classe não sadia foram geradas pelas operações cortar e duplicar das imagens já existentes, de modo que ambas as classes tenham a mesma quantidade de dados.

O modelo proposto por Baffa e Lattari (2018) obteve 98% de acurácia para o protocolo estático e 95% para o protocolo dinâmico. Considerando as estratégias para o protocolo dinâmico os melhores resultados foram obtidos pelas estratégias 1 e 3, ambos com 95% de acurácia. Como a primeira estratégia utiliza todas as imagens dinâmicas como dados de entrada, o modelo acaba ligando com mais dados redundantes. Por outro lado, como a estratégia 3 utiliza apenas a média da primeira e da última imagem, uma quantidade

menor de dados é necessária para entregar um modelo classificador, sendo assim, um modelo mais eficiente.

Roslidar et al. (2019) utilizaram *Convolutional Neural Networks* (CNNs) para classificar as imagens de termografia de mama do conjunto DMR. Eles executaram o balanceamento das bases, para ambos os protocolos, estático e dinâmico. Contudo não foi especificado o método utilizado. As CNNs utilizadas por Roslidar et al. (2019) foram: ResNet101, DenseNet, MobileNetV2 and ShuffleNetV2 pré-treinadas no conjunto de imagens ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) de acordo com o protocolo *fine-tuning* da transferência de aprendizagem. Nesta abordagem a camada completamente conectada da CNN é treinada para o conjunto DMR, enquanto os parâmetros das outras camadas são mantidos. Durante o treinamento da rede, o conjunto de dados foi agrupado em dois, 80% para dados de treinamento e 20% para validação de dados, as imagens apenas foram redimensionadas para $224 \times 244 \times 3$ antes do treinamento. Em seus resultados, a DenseNet201 foi capaz de classificar imagens estáticas e dinâmicas com 100% de acurácia. Destaque também para a MobileNetV2, considerada mais eficiente em tempo de treinamento, menor perda de treinamento e por alcançar 99,6% de acurácia.

Semelhante ao trabalho de Roslidar et al. (2019), Chaves et al. (2020) utilizaram CNNs para classificar imagens estáticas do conjunto DMR. Chaves et al. (2020) trabalharam com imagens de apenas 88 pacientes, sendo 44 pacientes sadias e 44 com câncer. Totalizando 440 imagens, com 220 imagens para cada classe. As CNNs aplicadas foram: AlexNet, GoogLeNet, ResNet18, VGG16 and VGG-19 pré-treinadas no ImageNet. Através do protocolo *fine-tuning* eles treinarem apenas as camadas completamente conectadas de cada rede. As redes que obtiveram os melhores resultados de acordo com o experimento de Chaves et al. (2020) foram as VGG16 e VGG19, ambas com 20 épocas. VGG16 atingiu 77,5% de acurácia, 90% de sensibilidade e 65% de especificidade, e VGG19 conseguiu 77,5% de acurácia, 90% de sensibilidade e 65% de especificidade.

Ekici e Jawzal (2020) propuseram uma CNN com parâmetros otimizados através do algoritmo de otimização de Bayes. Eles também executaram o balanceamento de base ao realizar diferentes tipos de conversões como translação, simetrias e rotação das imagens originais da classe minoritaria. Ao todo, eles trabalharam com 140 imagens entre sadias e não sadias da base DMR. Os primeiros resultados obtidos por Ekici e Jawzal (2020) atingiram 97,91% de acurácia, após a otimização dos parâmetros eles conseguiram 98,95%

de acurácia.

Mishra et al. (2020) trabalharam com imagens dinâmicas disponíveis do conjunto DMR (SILVA et al., 2014). Eles trabalharam com 521 imagens classificadas como ”Sadias” e 160 com ”Problema” (imagens que possui regiões com crescimento cancerígeno). Todas as imagens foram convertidas em níveis de cinza e redimensionadas para garantir que todas tivessem o mesmo tamanho. Para balancear a base de dados, eles usaram o aumento em tempo real usando rotação aleatória, translação X e translação Y. Em seguida, as imagens foram segmentadas e classificadas utilizando um modelo de CNN. Eles atingiram 95,5% de acurácia.

Utilizando um modelo InceptionV3 pré-treinado no ImageNet, Mambou et al. (2018) classificaram imagens frontais dinâmicas do conjunto DMR. Ao todo, eles utilizaram imagens de 67 pacientes, sendo 43 saudáveis e 24 doentes. No modelo proposto por eles, um classificador SVM pode ser utilizado se a saída da CNN for incerta. Isto é, se a confiança de saída da InceptionV3 para a classe ”doente” é maior que 0,5 e menor do que 0,6, é utilizado uma SVM para classificar a imagem utilizando os atributos extraídos pela CNN. As imagens utilizadas foram convertidas a níveis de cinza. Além disso, a região de interesse foi extraída removendo regiões indesejadas, como braços e região do pescoço. O método proposto por Mambou et al. (2018) atingiu resultados com área sob a curva ROC (AUC) igual a 1.

3.2 Trabalhos de pós-graduação relacionados produzidos na UFPE

Este trabalho é fruto do projeto de pesquisa realizado na UFPE denominado “Análise da viabilidade do uso de câmera termográfica como ferramenta auxiliar no diagnóstico de câncer de mama em hospital público localizado em clima tropical”. A partir desse projeto uma série de outras pesquisas sobre a termografia de mama vêm sendo realizadas, desde o projeto de aquisição das imagens termográficas e construção da base de imagens, como também projetos que visam o estudo de propriedades termofísicas dos tecidos da mama e também os projetos que avaliam a capacidade discriminatória de ferramentas computacionais na avaliação de imagens de termografia de mama, como esta tese de doutorado. Esses trabalhos foram realizados por pesquisadores tanto do departamento de Engenharia Mecânica quanto do departamento de Engenharia Biomédica da UFPE. Assim, os trabalhos citados a seguir são alguns dos projetos realizados nesse contexto dos

Programas de Pós-Graduação em Engenharia Mecânica e em Engenharia Biomédica da UFPE.

Entre os trabalhos realizados por pesquisadores da Engenharia Mecânica, vale destacar o trabalho realizado por Oliveira (2012), o qual descreve o protocolo para a padronização da aquisição das imagens de termografia, como descrito na Seção 2.1. Contudo antes do trabalho de Oliveira (2012), BEZERRA (2007) avaliou a aplicabilidade da termografia na detecção do câncer de mama. Para isto, BEZERRA (2007) fez a comparação entre temperaturas obtidas por simulações numéricas e das temperaturas medidas pela câmera termográficas. As análises numéricas foram feitas através de um modelos bidimensional e tridimensional. O modelo bidimensional não entregou resultados satisfatórios, contudo, apesar de uma geometria simplificada, o modelo tridimensional forneceu temperaturas próximas das temperaturas máximas dos tumores. Ainda antes da definição do protocolo, Araújo (2009) desenvolveu um sistema de banco de dados capaz de armazenar informações importantes sobre pacientes, inclusive imagens de termografia. Sendo possível, a partir do banco de dados o cruzamento de informações sobre cada paciente.

SILVA (2015) trabalhou com a classificação de imagens de termografia de mama adquiridas tanto em ambientes sem um controle adequado das variáveis térmicas como em ambientes controlados, seguindo o protocolo de Oliveira (2012). O objetivo do estudo foi avaliar se a termografia pode ser utilizada para triagem de lesões mamárias em regiões de poucos recursos. Para a classificação das imagens SILVA (2015) realizou experimentos considerando três tipos de segmentação de imagens e dois classificadores, o Discriminante Linear e um classificador baseado em distâncias. Considerando seus resultados, SILVA (2015) concluiu que a termografia pode ser utilizada para triagem em regiões de poucos recursos.

O trabalho desenvolvido por QUEIROZ (2016) teve como o objetivo o desenvolvimento de uma *Graphical User Interface* (GUI) de fácil uso e que possibilite a detecção de anormalidades de lesões mamárias. Para isto, a avaliação das imagens do sistema é feita através de segmentações automáticas e semiautomáticas, e em seguida as imagens são submetidas a classificadores, onde os atributos considerados foram baseados em medidas estatísticas e intervalares. Quanto a essa tarefa, QUEIROZ (2016) avaliou o desempenho da Máquina de Vetor de Suporte e Mahalanobis, um classificador baseado em distâncias.

Vasconcelos (2017) analisou métodos de aprendizagem de máquina na classificação de imagens de termografia para a detecção de lesões mamárias. Para isso, ela trabalhou com duas metodologias diferentes. Na primeira metodologia de Vasconcelos (2017), a sequência de passos foi a seguinte: Aquisição de Imagens; Balanceamento do conjunto de imagens por escolha aleatória; Segmentação automática; Extração de atributos; Seleção de atributos; Classificação. Já a segunda metodologia foi realizada seguindo tais passos: Aquisição de imagens; Balanceamento por vetores sintéticos; Segmentação automática; Extração de atributos; Classificação. A extração de atributos das imagens foi realizada através de medidas estatísticas e em medidas intervalares dos valores de temperatura. Vasconcelos (2017) executou duas abordagens em seus experimentos. Na primeira foi feita uma classificação binária Câncer x Não-Câncer. Onde a classe Não-Câncer era composta por imagens de pacientes sem lesão, com lesão benigna ou com cisto. Enquanto que na segunda abordagem, foi feita uma classificação multiclasse com imagens de pacientes com lesão maligna, com lesão benigna, com cisto e sem lesão. Ao todo foram considerandos os seguintes classificadores: *Naïve Bayes, Bayes Net*, Perceptron de Multicamadas, *Random Forest*, *Random Tree*, KNN (*K-nearest neighbours*) e Máquina de Vetor de Suporte. Para a primeira abordagem foram obtidos resultados com 93,42% de acurácia, 94,73% de sensibilidade e 92,10% de especificidade para a Classe Câncer. Já para a segunda 63,46% de acurácia, 80,77% de sensibilidade e 86,54% de especificidade para a Classe Maligno, ambos os melhores resultados foram atingidos com a Máquina de Vetor de Suporte.

Utilizando o mapeamento das temperaturas superficiais da mama de imagens de termografia ESPINDOLA (2017) estimou a condutividade térmica e perfusão sanguínea da mama e suas anomalias através da análise de uma paciente portadora de uma lesão maligna e também de um fantoma de dorso feminino. As simulações numéricas foram feitas usando o Método Determinístico de Programação Quadrática Sequencial (SQP) considerando um método inverso e a Equação da Biotransferência de Calor. Através de seus resultados, ESPINDOLA (2017) não recomenda a utilização do SQP para a estimação de parâmetros termofísicos da mama utilizando imagens termográficas. Pois, os valores obtidos dos parâmetros podem não ser do mínimo global da função objetivo do problema de otimização considerando.

MELO et al. (2019) apresentou uma metodologia, baseada em imagens de termografia, para o desenvolvimento de um modelo tridimensional da mama personalizado para

cada paciente. O objetivo de tal modelo é prover a geometria necessária para simulações numéricas com a finalidade de estimar propriedades termofísicas dos tecidos da mama. O modelo proposto por MELO et al. (2019) foi validado através da estimava das condutividades térmicas dos tecidos glandular, adiposo e de lesões utilizando Programação Sequencial Quadrática (SQP), onde os valores obtidos foram comparados com trabalhos anteriores e valores medidos pela câmera térmica.

Entre os trabalhos realizados por pesquisadores da Engenharia Biomédica, vale a pena destacar o realizado por SILVA (2019) na classificação de imagens de termografia de mama utilizando como extratores de atributos o extrator de textura de Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973) e os descritores de forma de Zernike. Em seu trabalho SILVA (2019) procedeu uma classificação única entre as imagens com cisto, lesão benigna, lesão maligna e sem lesão. Diferente deste trabalho que realiza duas classificações um entre as imagens com e sem lesão e outra entre as imagens com cisto, lesão benigna e lesão maligna. A maior contribuição do trabalho de SILVA (2019) foi a aplicação de métodos de seleção de atributos baseados em Algoritmo Genético (AG) e Otimização por Enxame de Partículas (OEP) para reduzir o número de atributos da base de dados que continha originalmente 169 atributos. Os série de classificadores foram utilizados para avaliar o desempenho dos subconjuntos selecionados, sendo os melhores resultados obtidos pela Máquina de Vetor de Suporte. No seu estudo foi possível reduzir o número de atributos para valores entre 57 com o AG e 60 com a OEP. Apesar da redução de atributos, não houve uma queda significativa no desempenho da classificação quando comparado com os testes realizados com conjuntos que possuíam todos os atributos.

No trabalho de SANTANA (2020) é investigado abordagens para um sistema de classificações de imagens de mamografia e termografia de mama. As imagens de termografia utilizada foram as adquiridas pela UFPE, enquanto que as imagens de mamografia analisadas foram da base IRMA. SANTANA (2020) realizou a extração de atributos das bases de imagens de duas maneiras. Na primeira foram utilizados os momentos de Haralick e Zernike, enquanto na segunda os atributos foram obtidos pela *Deep-Wavelet Neural Network* (DWNN). A partir disso, extensivos experimentos foram realizados utilizando diversos classificadores como *Bayes Net*, *Naïve Bayes*, Perceptron de Multicamadas, Máquina de Aprendizagem Extremo, árvore de decisão, Máquina de Vetor de Suporte, *Random Tree* e *Random Forest*. Os melhores resultados foram obtidos com as bases obtidas pela DWNN

quando classificadas pela Máquina de Vetor de Suporte.

4 METODOLOGIA: CLASSIFICAÇÃO DE IMAGENS DE TERMOGRAFIA UTILIZANDO *DEEP-WAVELET NEURAL NETWORK*

Neste trabalho é investigado o uso de técnicas de aprendizagem profunda para extração de atributos de imagens termográficas de mama. Também é proposto um método de extração de atributos baseado na decomposição *wavelets*, denominado *Deep-Wavelet Neural Network* (DWNN). A formulação matemática da DWNN está presente na Seção 4.1. A DWNN é aplicada ao problema de classificação de imagens de termografia de mama e o seu é comparado com os resultados de seis redes neurais convolucionais (CNNs) do estado da arte. As CNNs utilizadas neste trabalho foram pré-treinadas no conjunto de dados ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Essa é uma abordagem da aprendizagem por transferência onde as últimas camadas totalmente conectadas das redes são desconsideradas. Assim, as saídas das CNNs são obtidas pelas suas últimas camadas convolucionais. Os mapas das características gerados para os diferentes extratores de atributos foram classificados por máquinas de vetores de suporte (SVM) e máquinas de aprendizado extremo (ELM). Os detalhes referente a execução dos experimentos do trabalho estão na Seção 4.2.

4.1 Deep-Wavelet Neural Network

Esta seção é dedicada ao desenvolvimento de uma das contribuições deste trabalho de doutorado: a formulação matemática da *Deep-Wavelet Neural Network* (DWNN). A *Deep-Wavelet Neural Network* é um método da aprendizagem profunda de extração de atributos para o reconhecimento de padrões, baseado no algoritmo de Mallat (MALLAT, 1989) para a decomposição *wavelet* em múltiplos níveis.

Na decomposição *wavelet* baseada no algoritmo de Mallat filtros passa-baixa e passa-alta são aplicados a uma imagem, resultando em um conjunto de outras imagens. Imagens resultantes dos filtros passa-baixa e passa-alta são chamadas de aproximações e detalhes, respectivamente (MALLAT, 1989). Nas aproximações são destacadas as suavidades da imagem original, enquanto nos detalhes são destacados as bordas (ou regiões de descontinuidade). A decomposição *wavelet* é baseada em níveis. No primeiro nível a imagem de entrada é decomposta em uma imagem de aproximação e três imagens

de detalhes. Cada imagem de detalhe é resultado da aplicação de filtros passa-alta que destacam, cada um, bordas horizontais, verticais e diagonais. A Figura 44 mostra o primeiro nível da decomposição *wavelet* para uma imagem. No segundo nível, cada imagem de detalhe é avaliada de acordo com os mesmos filtros passa-baixa e passa-alta do nível um. A imagem de aproximação do nível um não é utilizada no nível subsequente. Esse processo pode seguir dessa maneira para o nível três em diante. Tal estratégia é utilizada no reconhecimento de padrões ao permitir analisar imagens tanto no domínio espacial quanto o domínio da frequência (MALLAT, 1989).

Na abordagem da DWNN, um neurônio é formado pela combinação de um dado filtro com a operação de *downsampling*, como é mostrado na Figura 45, onde as matrizes **X** e **Y** em destaque representam uma imagem digital.

Todos os filtros utilizados na DWNN formam um banco de filtros, que são mantidos fixos durante todo o processo. Supõe-se que o banco possua n filtros. Sendo assim, uma imagem de entrada será submetida a n neurônios que formam a primeira camada intermediária da rede neural. Na segunda camada, as imagens resultantes da primeira serão submetidas ao mesmo banco de filtros e ao *downsampling* individualmente, da mesma forma como foi feito para a imagem de entrada. E o processo se repete para a terceira e para as subsequentes camadas intermediárias. Por fim, na camada de saída da DWNN, se tem o bloco de síntese o qual é responsável por extrair informações das imagens resultantes do processo. Tal abordagem é esquematizada na Figura 46, na figura a sigla ”BS” significa bloco de síntese, e m representa o número de camadas da DWNN. O banco de filtros, o *downsampling* e o bloco de síntese serão detalhados nas seções a seguir.

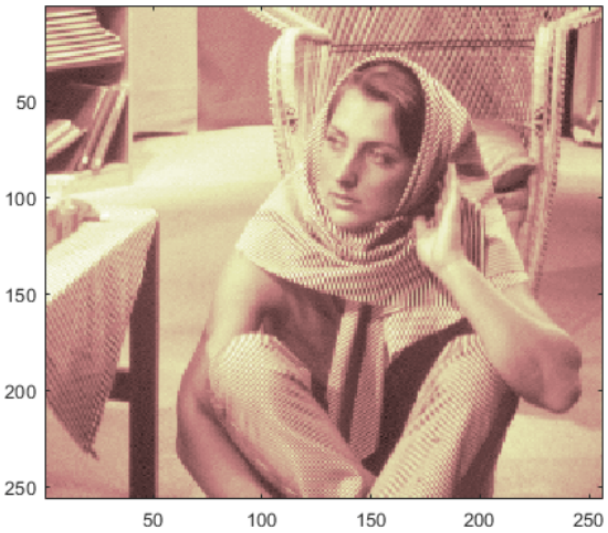
4.1.1 Banco de filtros

O banco de filtros utilizado na DWNN é fixo e composto por filtros ortogonais. Considerando S o domínio da imagem (chamada de suporte) e $\mathbb{R}$ o conjunto dos números reais, pode-se afirmar que os filtros ortogonais são do tipo $g_k : S \to \mathbb{R}$, para $1 \leq k \leq n$. Então, matematicamente, o banco de filtros (G), pode ser representado por:

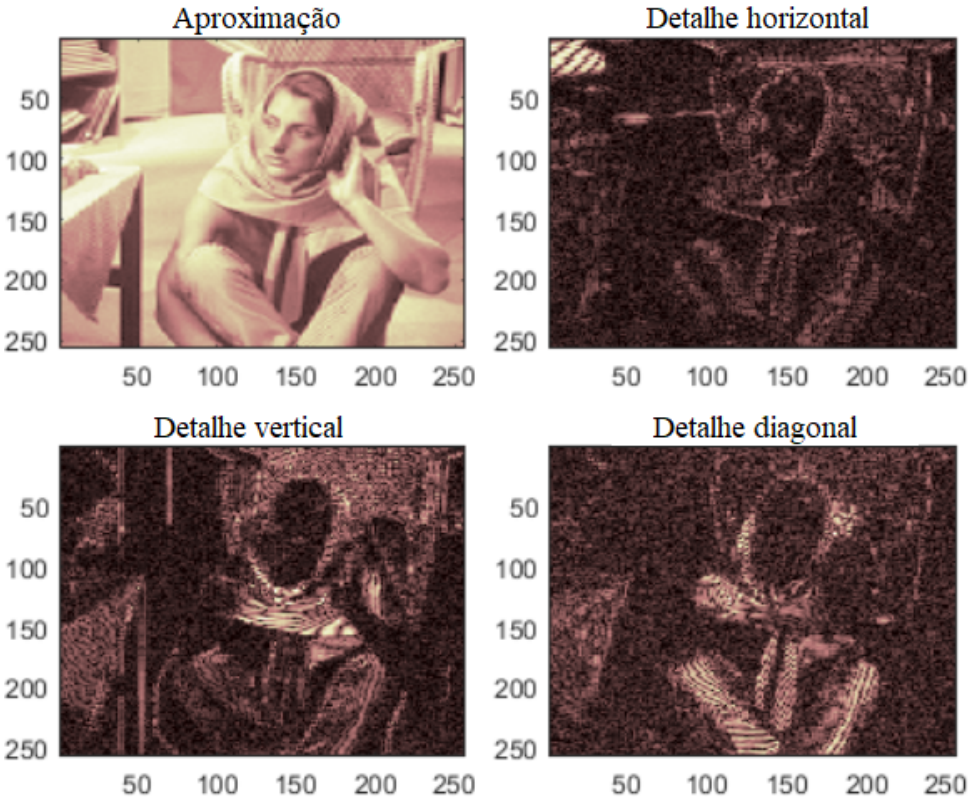
$$G = \{g_1,\ g_2,\ g_3,\ \ldots,\ g_n\} \tag{4.1}$$

Figura 44 – Primeiro nível da decomposição *wavelet*.

(a) Imagem de entrada

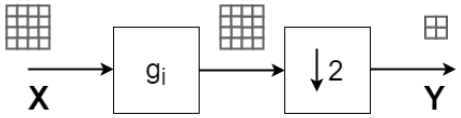


(b) Imagens de aproximação e detalhe.



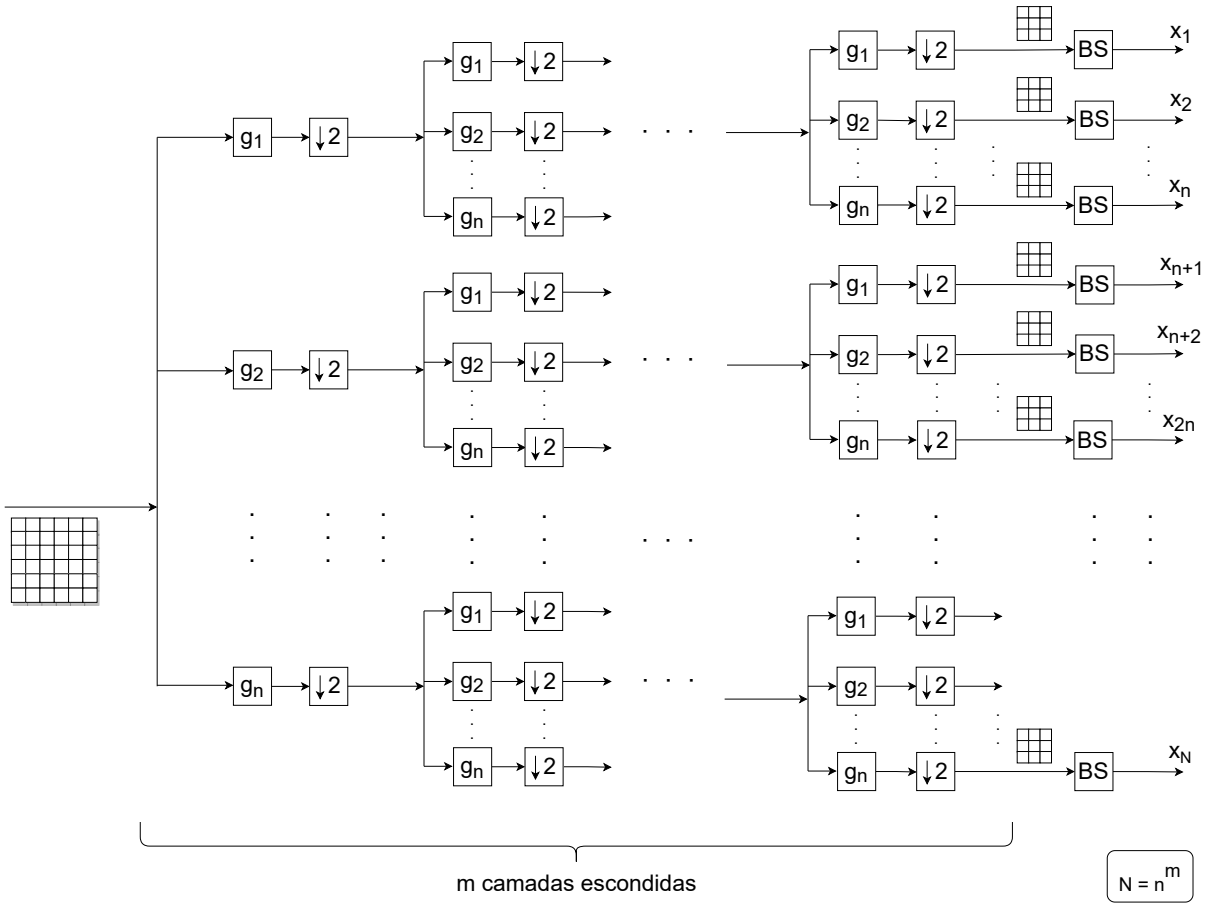
Fonte: Adaptado de MathWorks (2021).

Figura 45 – Neurônio da DWNN, g_i representa um filtro qualquer e ↓2 representa o *downsampling*. X e Y são imagens de entrada e saída do neurônio.



Fonte: O Autor (2022).

Figura 46 – Esquematização da abordagem da *Deep-Wavelet Neural Network* (DWNN).



Fonte: O Autor (2022).

Antes de determinar o banco de filtros da DWNN, é necessário primeiro definir qual vizinhança será considerada durante o processo de filtragem. A definição dos filtros g_k é feita apenas após a definição da vizinhança.

Seja um *pixel* $\vec{u} = (i, j)$, onde (i, j) representa as coordenadas do *pixel* $\vec{u}$. A vizinhança de 8 *pixels* (também chamada de vizinhança-8) de $\vec{u}$ é formada pelos *pixels*: $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, $(i, j-1)$, $(i+1, j+1)$, $(i+1, j-1)$, $(i-1, j+1)$ e $(i-1, j-1)$. Ou seja, são considerados como vizinhos os *pixels* laterais, verticais e diagonais de $\vec{u}$, isto é mostrado na Figura 47a.

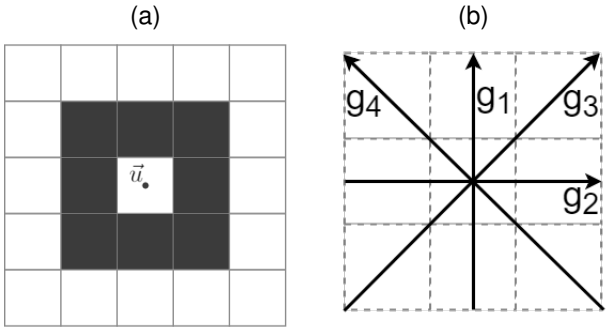
Considerando uma vizinhança-8, é possível formar uma base ortonormal de filtros contendo ao todo cinco filtros, onde quatro filtros são filtros passa-banda contendo uma seletividade de orientação específica (MALLAT, 1989). Ou seja, cada filtro irá destacar detalhes em uma dada orientação. Na Figura 47b são mostradas as orientações de tais filtros para uma vizinhança-8. Na imagem, g_1 é o filtro vertical de alta frequência, responsável

por destacar bordas horizontais, g_2, filtro horizontal de alta frequência, que destaca bordas verticais, e g_3 e g_4 são os filtros diagonais, os quais destacam as quinas da imagem.

Assim os filtros g_1, g_2, g_3 e g_4 formam o conjunto de filtros passa-alta, também classificados como filtros derivativos por destacar as descontinuidades da imagem de entrada.

Além dos filtros passa-alta, o banco de filtros da DWNN conta com mais um filtro, um passa-baixa (g_5), que atua como um suavizador, tendo como função destacar as áreas homogêneas da imagem. Tal filtro é considerado como um filtro integrador. Um exemplo de filtro passa-baixa normal para a vizinhança-8 é dada na Equação 4.2 onde o centro da máscara (2,2) é o *pixel* a ser substituído durante a filtragem.

Figura 47 – (a) Vizinhança-8, são considerados vizinhos do *pixel* $\vec{u}$ todos os *pixels* marcados em cinza. (b) Filtros passa-alta com seletividade de orientação, g_1 filtro com seletividade vertical, g_2, filtro horizontal, g_3 e g_4 filtros diagonais.



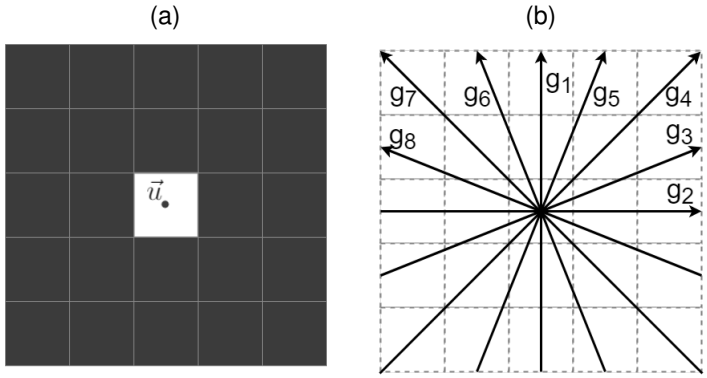
Fonte: O Autor (2022).

$$g_5 = \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{(2,2)} \tag{4.2}$$

Os filtros mostrados pela Figura 47b e pela Equação 4.2 são válidos para uma vizinhança-8. Contudo, ao optar por uma vizinhança diferente devem-se considerar outros filtros, por exemplo, considerando-se uma vizinhança de 24 *pixels* como mostra a Figura 48a. Os filtros passa-alta de seletividade de orientação específica serão dados como os mostrados na Figura 48b. Neste caso, há 8 filtros passa-alta ($g_1,\ g_2,\ \ldots,\ g_8$). Para a vizinhança-24, o filtro passa-baixa, seria uma matriz 5×5 formada por termos $1/25$, seguindo o mesmo princípio do filtro dado na Equação 4.2 para uma vizinhança-8. Então, para uma vizinhança de 24 *pixels* deve-se usar como banco de filtros na DWNN o conjunto

ortonormal dados pelos filtros mostrados na Figura 48b e mais um passa-baixa, totalizando 9 filtros.

Figura 48 – (a) Vizinhança para 24 *pixels*. (b) Filtros passa-alta com seletividade de orientação para uma vizinhança de 24 *pixels*.



Fonte: O Autor (2022).

Como mostrado no esquema da Figura 46, os dados resultantes de todos os filtros da DWNN são considerados na camada subsequente. Assim, na DWNN, as imagens resultantes da filtragem com o filtro passa-baixa são propagadas nas camadas posteriores. Diferente da abordagem convencional da decomposição *wavelet*, onde as imagens de aproximação não são utilizadas nos níveis sequentes.

Na decomposição de imagens em níveis de aproximação e detalhes, como na decomposição *wavelet* e na DWNN, as máscaras pequenas destacam deformidades pequenas. Assim, nas primeiras decomposições, são destacadas pequenas alterações da imagem de entrada. Porém à medida que esse processo avança pelas camadas, essas alterações vão se tornando borradas e são incorporadas ao fundo, restando apenas as deformidades grandes da imagem. Assim à medida que as imagens vão sendo embaçadas, as camadas finais são responsáveis por destacar as grandes deformidades. Essa característica, torna essa abordagem adequada para problemas com imagens de termografia de mama, pois no contexto de lesões mamárias, os achados podem ser tanto de pequeno porte quanto de grande porte.

4.1.2 *Downsampling*

O segundo processo componente de um neurônio da DWNN é o *downsampling*, responsável pela redução do tamanho da imagem. O uso do *downsampling* possui uma característica interessante ao diminuir o consumo de memória durante a execução do algo-

ritmo. Considere, por exemplo, uma imagem com 4096 *pixels* submetida ao banco de filtros ortonormais de vizinhança-8 (referente à primeira camada intermediária da DWNN). Como resultado serão obtidas outras $n = 5$ imagens cada uma contendo a mesma quantidade de *pixels*. Resultando, então, em um aumento da quantidade de dados por um fator de cinco. Ao serem consideradas mais camadas da DWNN, a quantidade de dados cresceria exponencialmente por um fator n^m.

Tal inconveniente poderia inviabilizar o processo. Mas ao aplicar o *downsampling*, a medida que o número de imagens aumenta por um fator n^m, o tamanho de cada imagem proveniente do processo é reduzido com relação a imagem de entrada por um fator 4^{-m}. Uma situação especialmente interessante ocorre ao ser considerada uma vizinhança-8, para qual o banco de filtros possui ao todo 5 filtros. Entretanto, é possível combinar os filtros diagonais (g_3 e g_4 na Figura 47b) de modo a trabalhar com um banco contendo 4 filtros. Nesse caso especial, após m camadas de neurônios da DWNN serão obtidas 4^m imagens reduzidas cada uma, com relação a imagem de entrada, por um fator de 4^{-m}. Dessa forma, então, a quantidade de dados se manterá constante durante toda a execução do algoritmo. Ao se utilizar tal abordagem na imagem de 4096 *pixels* do exemplo anterior, após a primeira camada da DWNN, ter-se-á como resultado 4 imagens de 1024 *pixels*. Permanecendo constante, então, a quantidade de dados durante o processo.

Considerando o esquemático de um neurônio da DWNN mostrado na Figura 45 e as imagens de entrada **X** e de saída **Y** para um dado neurônio, pode-se escrever que **Y** está relacionada com **X** da seguinte forma:

$$\mathbf{Y} := \phi_{\downarrow 2}(g_k * \mathbf{X}) \tag{4.3}$$

onde o símbolo $*$ representa a operação de convolução.

4.1.3 Bloco de síntese

A camada de saída da DWNN é formada pelos blocos de síntese. Cada bloco tem por função extrair, de cada imagem resultante das camadas intermediárias, uma informação (ou dado) que a represente, como é mostrado na Figura 49.

Então, nos blocos de síntese cada uma das n^m imagens reduzidas serão submetidas

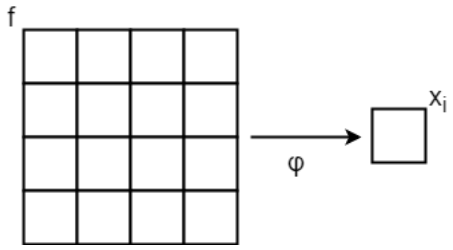
a uma função $\varphi : S \rightarrow \mathbb{R}$. Entre outras possibilidades, $\varphi(\cdot)$ pode assumir uma função de máximo, de mínimo, média ou mediana. Seu objetivo é, assim, substituir toda a imagem por um único valor.

Dessa forma, considerando $f(\vec{u}) \in \mathbb{R}$ o valor do *pixel* $\vec{u}$, tem-se que $x_i \in \mathbb{R}$ é obtido da seguinte maneira:

$$x_i = \varphi(f(\vec{u});\ \forall \vec{u} \in S) \tag{4.4}$$

Na Figura 49 é mostrado um esquemático do bloco de síntese. Ao final da DWNN, ao aplicar o bloco de síntese a todas as imagens resultantes das m camadas intermediárias, obtém-se um conjunto de termos x_i ($1 \leq i \leq n^m$). Tal conjunto pode ser entendido como os atributos da imagem de entrada. Ao se aplicar a DWNN a um conjunto de imagens, obtém-se como resultado um banco de dados, que pode ser utilizado como entrada de um classificador.

Figura 49 – Esquematização do processo de síntese



Fonte: O Autor (2022).

Na Tabela 9 são mostrados os números de parâmetros treináveis (em milhões) da DWNN com duas, quatro e seis camadas e para seis CNNs do estado da arte. A partir da tabela é possível analisar que a DWNN é um extrator de atributos simples e sem parâmetros de treinamento. Ao contrário das CNNs convencionais que precisam milhões de parâmetros para serem treinados.

4.2 Análise das imagens de termografia

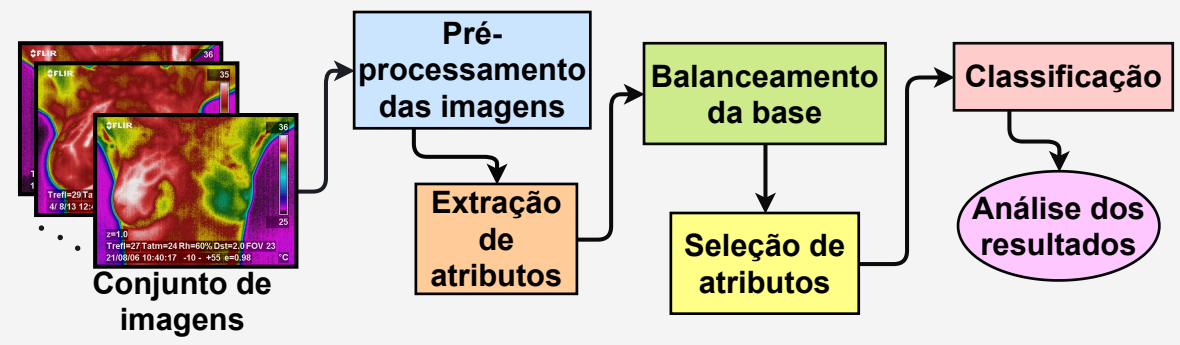
Os experimentos realizados nesse trabalho seguiram as etapas destacadas no fluxograma dado na Figura 50. As etapas dos experimentos são descritas nas subseções a seguir.

Tabela 9 – Número de parâmetros treináveis (em milhões) da DWNN com duas, quatro e seis camadas e para seis CNNs do estado da arte.

Modelo	Parâmetros (em milhões)
DWNN2	0
DWNN4	0
DWNN6	0
InceptionV3	23,9
MobileNet	4,2
ResNet50	25,6
VGG16	138
VGG19	144
Xception	22,9

Fontes: Simonyan e Zisserman (2015), Szegedy et al. (2016), He et al. (2016), Chollet (2017), Howard et al. (2017), Chollet (2015).

Figura 50 – Visão geral do método proposto: as imagens térmicas da mama são selecionadas do conjunto de dados gerais e agrupadas de acordo com as seguintes estratégias: (a) detecção de lesões, em que as imagens são classificadas como saudáveis e não saudáveis; e (b) classificação das lesões, caracterizada pela classificação das imagens em cisto, lesão benigna e maligna. Os atributos são extraídos pelas arquiteturas de redes neurais profundas. Posteriormente, as classes são balanceadas usando vetores sintéticos obtidos pelo algoritmo SMOTE. A dimensão do vetor de características é reduzida pela seleção dos atributos mais relevantes através da *Random Forest*. Por fim, a classificação é realizada por SVMs e ELMs avaliados em vários *kernels*.



Fonte: O Autor (2022).

4.2.1 Conjunto de imagens

Esse trabalho utilizou as imagens obtidas pelo Departamento de Engenharia Mecânica da UFPE seguindo o protocolo descrito na Seção 2.1. Entre as imagens obtidas pelo grupo foram consideradas apenas as imagens frontais de cada paciente obtidas a distância fixa. Ou seja, foram consideradas apenas as imagens T1 e T2 (como as das Figuras 8a e 8b). Optou-se por trabalhar dessa forma porque essas imagens permitem uma visualização clara de ambas as mamas. O que favorece a identificação da região de interesse. Além disso, apesar de todas as imagens possuírem o mapa de calor à direita e legendas no

canto inferior, essas informações não influenciam no processo de classificação das imagens de termografia por serem padronizadas e estabelecidas no mesmo local, não havendo aleatoriedade.

As imagens utilizadas no trabalho estão rotuladas em: Sem Lesão, Cisto, Lesão Benigna e Lesão Maligna. As quantidades de imagens frontais de cada classe são dadas na Tabela 10. A partir dessas imagens os experimentos foram realizados de acordo com duas abordagem. Na primeira abordagem, a imagens rotuladas como Cisto, Lesão Benigna e Lesão Maligna são agrupadas para formar o conjunto Com Lesão, totalizando 270 imagens (como mostrado na Tabela 10). Dessa forma, uma base de imagens é formada com as imagens rotuladas ”com lesão” e as imagens rotuladas como ”sem lesão”. Essa base de imagens é então utilizada para uma classificação binária com o objetivo de analisar e diferenciar as imagens sadias das imagens com algum tipo de lesão. O objetivo dessa abordagem é avaliar a habilidade dos algoritmos de detectar a existência de alguma lesão em imagens de termografia de mama, e por isso essa abordagem recebe o nome de ”detecção de lesão”. Na segunda abordagem, chamada de ”classificação de lesão”, são levadas em consideração apenas as imagens rotuladas como cisto, lesão benigna e lesão maligna, assim, ignorando as imagens sem nenhum tipo de lesão mamária. Do ponto de vista da aprendizagem de máquina, na segunda abordagem é realizada uma classificação multiclasse, devido ao fato de esse problema conter três classes distintas. Para a segunda abordagem o objetivo é investigar a habilidade dos algoritmos de diferenciar três lesões mamárias.

Tabela 10 – Quantidade de imagens de termografia de mama para as classes utilizadas neste trabalho. Para a avaliação o desempenho dos algoritmos em detectar lesões as imagens das classes Cisto, Lesão Benigna e Lesão Maligna são reunidas para formar a classe Com Lesão, totalizando 270 imagens.

Classes		N° de Imagens	Total
Com Lesão	Cisto	73	270
	Lesão Benigna	121	
	Lesão Maligna	76	
Sem Lesão	-	66	66
			336

Fonte: O Autor (2022).

4.2.2 Pré-processamento das imagens

Para a execução dos experimentos, em primeiro lugar, as imagens de termografia de mama são pré-processadas. Para os experimentos com as CNNs, as imagens são redimensionadas para a resolução adequada para cada CNN. Assim, as imagens a serem utilizadas pelas MobileNet, ResNet50, VGG16 e VGG19 são redimensionadas para a resolução 224×224. Enquanto que as imagens a serem utilizadas pelas Inception V3 e Xception são redimensionadas para 299×299. Esse redimensionamento foi realizado em Python versão 3.7 utilizando a biblioteca Keras/TensorFlow (CHOLLET, 2015).

Por outro lado, para os experimentos com a DWNN, as imagens são convertidas em níveis de cinza. As imagens de termografias são coloridas através de uma paleta de pseudo-cor. Onde tons mais avermelhados e esbranquiçados representam temperaturas mais quentes e os tons mais azulados e escuro representam as temperaturas mais frias. Deste modo, a conversão das imagens coloridas para imagens em tons de cinza foi realizada através da conversão do mapa de cores RGB-JET para níveis de cinza de 8 bits (com níveis de 0 a 255). Após a conversão, as regiões mais claras das imagens representam as temperaturas mais altas e as regiões mais escuras representam as temperaturas mais baixas da imagem. A conversão para níveis de cinza foi realizada utilizando o SID-Termo, um software de autoria do Grupo de Pesquisa em Computação Biomédica, UFPE.

4.2.3 Extração de atributos

Após o pré-processamento, os atributos de cada imagem são extraídos pela DWNN e por seis CNNs. Os experimentos com a DWNN foram realizados considerando as arquiteturas do método com duas, quatro e seis camadas escondidas. Essas arquiteturas são denominadas, nesse texto, como DWNN2, DWNN6 e DWNN6. Nesse trabalho, para todas as arquiteturas da DWNN a função utilizada para o *downsampling* e o bloco de síntese na DWNN foi a função de máximo. Se optou por tal função em razão de ser a mais comum entre as CNNs (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Além disso, para o problema de classificação de imagens de termografia de mama, a função de máximo é adequada pois a temperatura máxima é um parâmetro importante na análise das imagens, pois o desenvolvimento de lesões tumorais é responsável pelo aumento da temperatura da região.

As CNNs do estado da arte utilizadas no trabalho foram:

- Inception V3, a terceira versão da redes Inceptions;
- MobileNet, versão padrão, sem os multiplicadores de largura e resolução;
- ResNet50, rede ResNet com 50 camadas;
- VGG16 e VGG19, redes VGGs com 16 e 19 camadas, respectivamente;
- Xception.

A utilização das CNNs foi realizada no contexto da transferência de aprendizagem. Dessa maneira, foram utilizadas CNNs pré-treinadas no conjunto de imagens ImageNet. A abordagem da transferência de aprendizagem utilizada foi a de extração de atributos, onde as redes atuam como um extrator de atributos abstratos, pois tais informações não possuem um significado o qual seja humanamente interpretável. Na abordagem de extração de atributos as imagens são propagadas pela rede até uma camada preestabelecida. Nesse trabalho, para todas as CNNs utilizadas, as imagens foram propagadas até a última camada convolucional, isto é, desconsiderando apenas as camadas totalmente conectada das redes. Assim o número de atributos extraídos por cada CNN depende dos dados entregues por sua última camada convolucional para as camadas totalmente conectada. Por exemplo, a VGG16 entrega um tensor $7\times7\times512$ em sua última camada convolucional (como é mostrado na Figura 24). Assim, para a VGG16 esses dados podem ser colocados em um vetor linha com 25088 atributos. Seguindo esse procedimento, Inception V3, MobileNet, ResNet50, VGG16, VGG19 e Xception extraem 131072, 50176, 100352, 25088, 25088 e 204800 atributos, respectivamente. Esses valores são dados na Tabela 11.

Os experimentos com as CNNs foram feitos utilizando Python 3.7 e a biblioteca Keras/TensorFlow (CHOLLET, 2015) onde é possível fazer o *download* das CNNs pré-treinadas. A extração de atributos com a DWNN foi feita o SID-Termo.

Nessa etapa, também foram medidos os tempos de extração dos atributos das imagens pela DWNN e as CNNs. Assim, para cada modelo mediu-se o tempo de extração 30 vezes.

4.2.4 Balanceamento da base

Como mostrado na Tabela 10, a distribuição de imagens frontais da base utilizada neste trabalho é desbalanceada. Isto é, uma ou mais classes possuem um número consideravelmente maior do que as outras. Pela tabela é possível observar que a base de imagens possui 121 imagens de lesão benigna, enquanto que as outras classes possuem um número de imagens da ordem de setenta. Como trabalhar com bases desbalanceadas pode enviesar a classificação para a classe majoritária, foi realizado o balanceamento das bases no mapa das características com o algoritmo SMOTE (CHAWLA et al., 2002; BLAGUS; LUSA, 2013). Esta etapa foi executada em Python 3.7.

Assim, na proposta desta pesquisa, são criados vetores artificiais de acordo com suas posições no mapa de características obtidos pelos métodos de extração de atributos. Diferente dos trabalhos de Baffa e Lattari (2018), Ekici e Jawzal (2020) e Mishra et al. (2020) comentados no Capítulo 3, onde a sobreamostragem é feita ao criar imagens artificiais para o balanceamento das bases.

4.2.5 Seleção de atributos

Ao analisar a Tabela 11, é possível verificar que o número de atributos extraídos pelos métodos, é em sua maioria, na ordem de milhares. Alguns chegando até centenas de milhares atributos. Como essa quantidade de atributos extraídos é grande, foi feita uma seleção de atributos nos conjuntos de dados com muitos atributos para evitar problemas envolvendo a dimensão dos dados, como a maldição da dimensionalidade. Assim, não foi executada a seleção de atributos pelas bases construídas por DWNN2 e DWNN4, pois esses métodos não entregaram um número demasiadamente grande de atributos.

A seleção de atributos foi feita utilizando a *Random Forest* com 1000 árvores. A implementação dessa etapa foi feita em Python 3.7. O número de atributos selecionados para cada método são dados na Tabela 11.

4.2.6 Classificação

Os conjuntos de dados obtidas pela DWNN foram classificados usando SVMs e ELMs. Para a SVM, foram utilizados os *kernels* linear e RBF. Para o *kernel* RBF foi variado

Tabela 11 – Número de atributos extraídos por cada modelo e número de atributos mais relevantes selecionados utilizando *Random Forest*.

Modelo	Atributos Extraídos	Atributos Selecionados
DWNN2	16	-
DWNN4	256	-
DWNN6	4.096	294
InceptionV3	131.072	384
MobileNet	50.176	294
ResNet50	100.352	294
VGG16	25.088	294
VGG19	25.088	294
Xception	204.800	600

Fonte: O Autor (2022).

o parâmetro gama entre os valores 0,01; 0,25 e 0,50. Por outro lado, os experimentos com as ELMs foram executados com 500 neurônios na camada oculta com *kernels* polinomiais de grau de 1 a 5. Esses primeiros experimentos tiveram como objetivo encontrar uma boa configuração da DWNN para o problema. Para os conjuntos de dados gerados pela extração de atributos usando as CNNs foram realizadas as classificações usando SVM com *kernel* linear. Os experimentos de classificação utilizando SVMs foram realizados no *software* Weka, versão 3.8.5 (HALL et al., 2009; BOUCKAERT et al., 2016), enquanto que os experimentos com as ELMs foram executados no SID-Termo.

Como se tem um número reduzido de imagens, os experimentos foram realizados com validação cruzada com 10 *fold*, assim todas as imagens foram utilizadas em um momento para treinamento, e em um outro momento para teste. Isso faz com que os conjuntos reservados para treino e teste dos classificadores não interfiram na acurácia da rede. O desempenho geral é dado pelo erro quadrático médio dos 10 testes.

Cada experimento considerado foram executados 30 vezes, por exemplo, o conjunto de dados extraídos pela DWNN6 foram classificados 30 vezes pela SVM com *kernel* linear. Assim, é possível analisar os resultados com significância estatística, onde é possível avaliar os resultados através de medidas como média e desvio padrão.

4.2.7 Análise dos resultados

Os resultados obtidos pela classificação das bases de dados foram avaliados pelas métricas discutidas na Seção 2.10, são elas: acurácia, índice Kappa, sensibilidade, especificidade e precisão. A análise dos dados foi feita utilizando Python 3.7.

4.3 Infraestrutura experimental

Para este trabalho foi utilizado o SID-Termo (software de autoria do Grupo de Pesquisa em Computação Biomédica, UFPE), Python na versão 3.7, Keras/TensorFlow (CHOLLET, 2015), e o Weka, versão 3.8.5 (HALL et al., 2009; BOUCKAERT et al., 2016). As finalidades de cada software estão descritas na seção anterior e estão resumidas na Tabela 12 de acordo com as etapas do fluxograma da Figura 50. Os experimentos foram executados em um CPU Intel Xeon Silver 4110 CPU @ 2,10 GHz, com 8 núcleos, 16 processadores lógicos e 128 GB de memória RAM. Os experimentos para a medição do tempo de execução dos algoritmos de extração de atributos, mostrados na Tabela 15, foram executados em um CPU Intel Core i5-4200U CPU @ 1,60 GHz, com 2 núcleos, 4 processadores lógicos e 6 GB de memória RAM.

Tabela 12 – Descrição da utilização dos softwares para este trabalho.

Etapa	Software	Função
Pré-processamento	SID-Termo	Conversão em níveis de cinza das imagens para aplicar a DWNN
	Keras/TensorFlow	Redimensionamento das imagens para aplicar as CNNs
Extração de atributos	SID-Termo	Extração de atributos com a DWNN
	Keras/TensorFlow	Extração de atributos com as CNNs pré-treinadas no ImageNet
Balanceamento da base	SID-Termo	Balanceamento para as bases da DWNN
	Python	Balanceamento para as bases das CNNs
Seleção de atributos	Python	Seleção de atributos usando *Random Forest*
Classificação	SID-Termo	Classificação usando ELM
	Weka	Classificação usando SVM
Análise dos resultados	Python	Construção dos boxplots.

Fonte: O Autor (2022).

5 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos pela classificação de imagens de termografia de mama para a detecção do câncer de mama. Os experimentos foram executados de acordo com duas abordagens: ”detecção de lesão” e ”classificação de lesão”. Na abordagem ”detecção de lesão” é feita a classificação binária entre as imagens das classes ”com lesões” (não saudáveis) e ”sem lesões” (saudáveis). Para isso, as imagens rotuladas como cisto, lesão benigna e lesão maligna são agrupadas na classe ”com lesões” conforme é mostrado na Tabela 10. A segunda abordagem é a ”classificação das lesões”, o qual é um problema de classificação com três classes: cisto, lesão benigna e lesão maligna. Desta forma, na segunda abordagem, é ignorado as imagens saudáveis do conjunto de dados. Cada configuração dos experimentos foram executadas 30 vezes. Assim, é possível analisar os resultados com significância estatística. Além disso, como se tem um número reduzido de imagens, os experimentos foram realizados com a validação cruzada com 10 *fold*. Também são avaliados os tempos de extração de atributos para cada método de extração, e algumas matrizes de confusão obtidas pelos métodos.

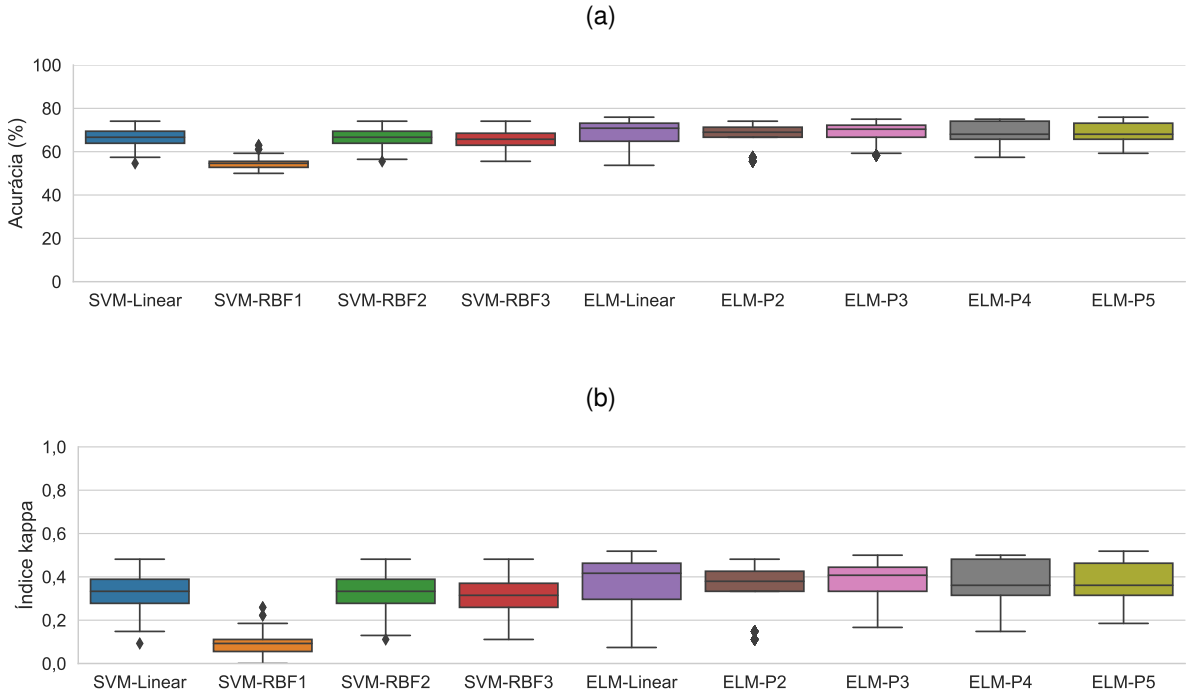
Os primeiros resultados a serem apresentados são os resultados das classificações utilizando conjuntos de dados obtidos da DWNN com diferentes números de camadas. Esses resultados serão analisados de acordo com a acurácia e o índice kappa, pois essas métricas avaliam o desempenho global das técnicas. Seja de acordo com as porcentagens de acerto das classificação das imagens, dados pela acurácia, ou de acordo com a concordância dos resultados obtidos, apresentada pelo índica kappa.

Nos primeiros experimentos, o número de camadas do método foi variado em dois, quatro e seis. A seguir, esses extratores serão denotados como DWNN2, DWNN4 e DWNN6, respectivamente. Quanto maior o número de camadas, maior o custo computacional do processo. No entanto, com mais camadas, mais atributos são extraídos de cada imagem. O número de atributos extraídos pelo modelo DWNN nos experimentos são iguais a 4^m, onde m é o número de camadas (como mostrado na Figura 46). Portanto, DWNN2, DWNN4 e DWNN6 fornecem 16, 256 e 4096 atributos, respectivamente. Para evitar a maldição da dimensionalidade, foi executada a seleção de atributos usando *Random Forest* para conjuntos de dados obtidos por DWNN6. O número de atributos extraídos e selecionados

são fornecidos na Tabela 11. Todos os experimentos foram executados com a função de máximo no operador de *downsampling* da DWNN.

As Figuras 51, 52 e 53 apresentam *boxplots* da acurácia e de índice kappa, para a abordagem de detecção de lesão (classificação binária de imagens sem lesão e com lesão), usando DWNN2, DWNN4 e DWNN6, respectivamente. Nessas figuras, SVM-Linear, SVM-RBF1, SVM-RBF2 e SVM-RBF3 representam os resultados usando SVM com *kernel* polinomial de grau um, e *kernel* RBF com gama 0,01; 0,25 e 0,50, respectivamente. Enquanto que ELM-Linear, ELM-P2, ELM-P3, ELM-P4 e ELM-P5 representam os resultados usando ELM com *kernel* polinomial de grau 1 a 5.

Figura 51 – Resultados de (a) acurácia e (b) índice kappa dos classificadores com a DWNN2 aplicados para a detecção de lesão.

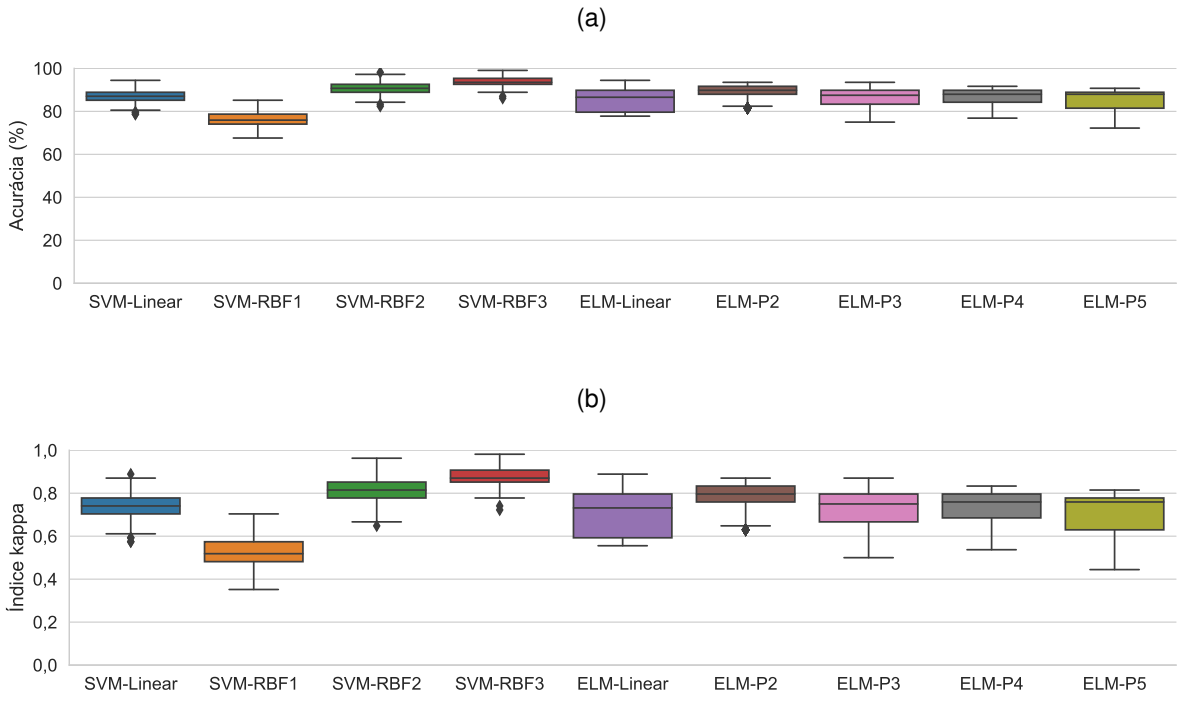


Fonte: O Autor (2022).

Os resultados considerando apenas duas camadas da DWNN podem ser considerados bastante limitados. No geral, os classificadores obtiveram cerca de 65% de acurácia. Já a maioria dos classificadores atingiram entre 0,2 e 0,4 para o índice kappa, indicando uma concordância razoável de acordo com a Tabela 8.

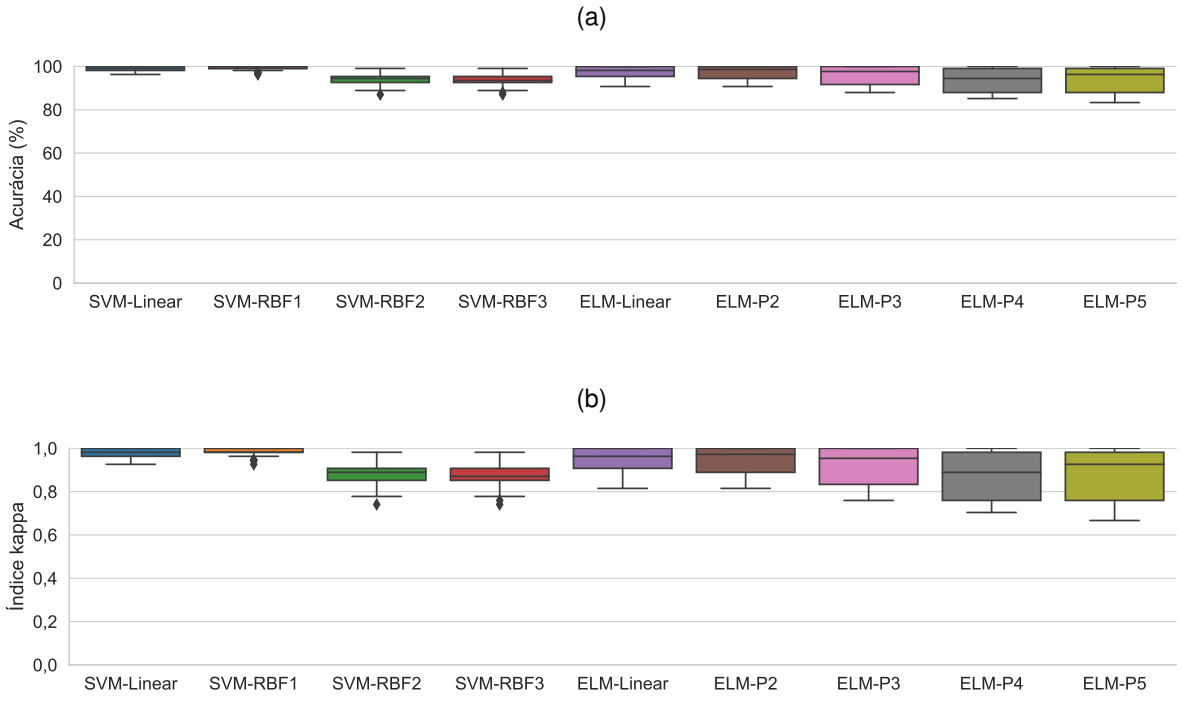
Melhores resultados foram obtidos com a DWNN4. Utilizando esse extrator de atributos, a maioria dos classificadores atingiram cerca de 85% de acurácia e 0,7 de índice kappa. No entanto, os melhores resultados foram obtidos com DWNN6. Com os atributos

Figura 52 – Resultados de (a) ac.urácia e (b) índice kappa dos classificadores com a DWNN4 aplicados para a detecção de lesão.



Fonte: O Autor (2022).

Figura 53 – Resultados de (a) acurácia e (b) índice kappa dos classificadores com a DWNN6 aplicados para a detecção de lesão.

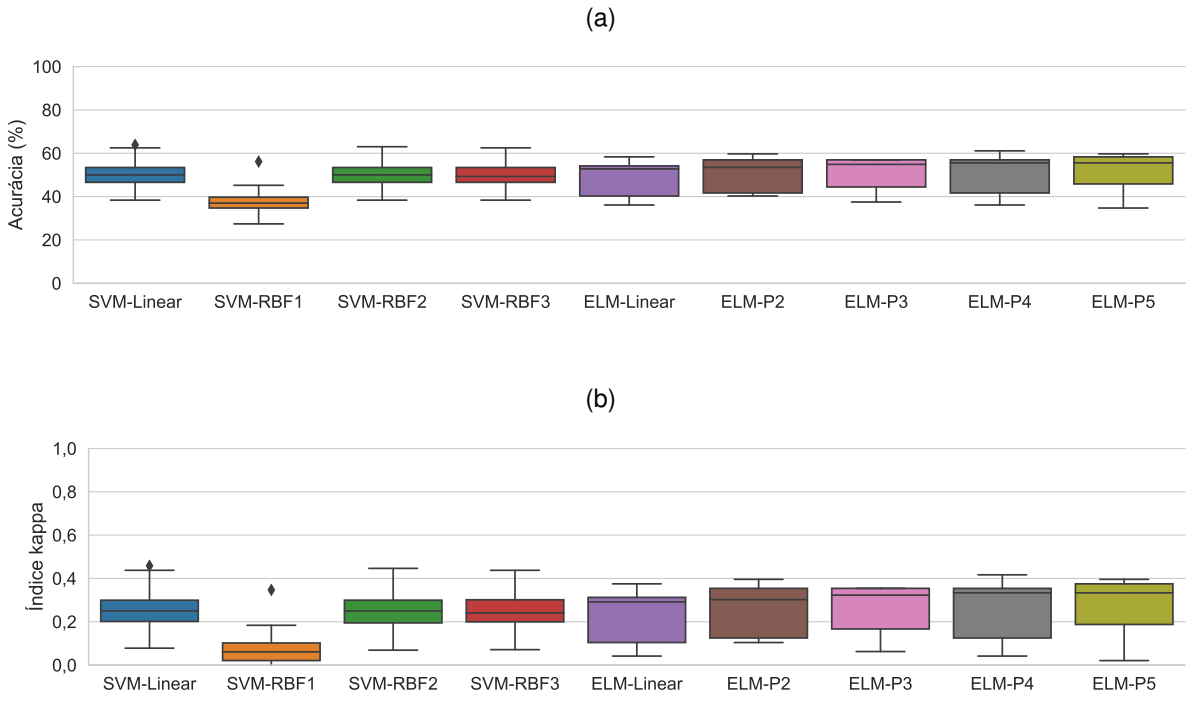


Fonte: O Autor (2022).

extraídos pela DWNN6, o SVM-Linear e o SVM-RBF1 obtiveram-se resultados próximos a 100% e 1,0 de acurácia e índice kappa, respectivamente. Bons resultados também foram alcançados com ELM, mas com resultados mais dispersos. Os resultados com menor dispersão obtidos pela SVM-Linear e pela SVM-RBF1 indicam que esses métodos possuem uma maior confiabilidade com relação às ELMs.

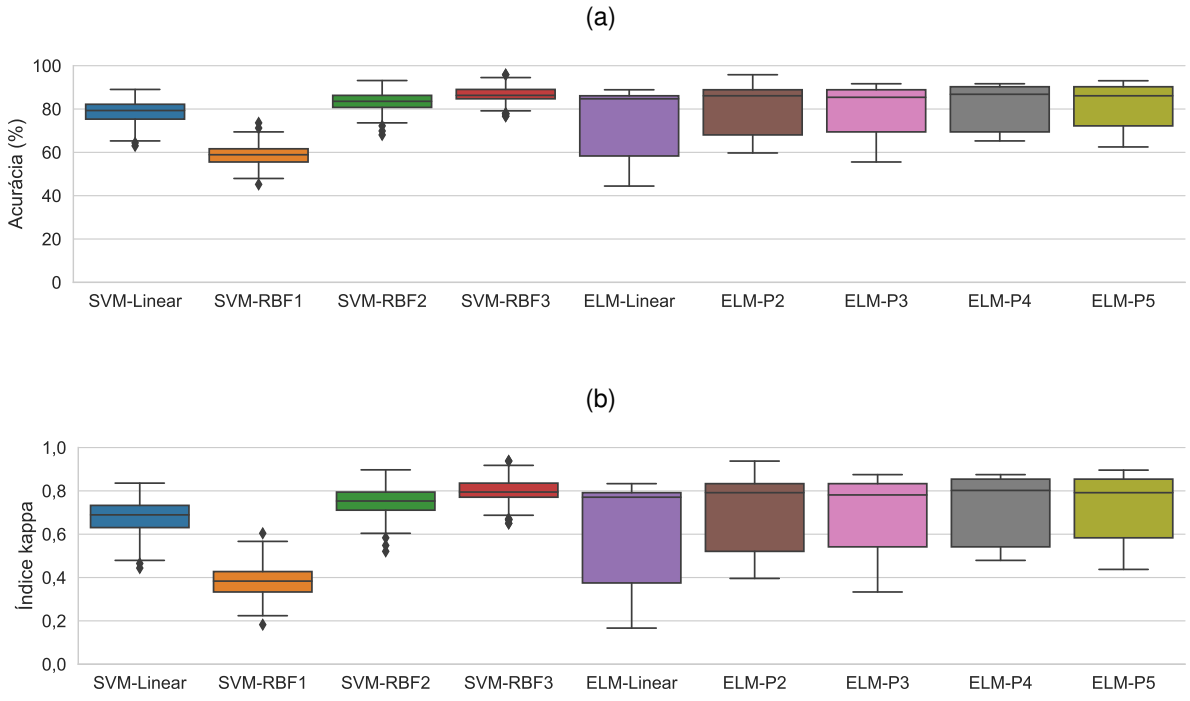
Seguindo a mesma ideia dos resultados para o problema de detecção de lesões, as Figuras 54, 55 e 56 apresentam resultados para o problema de classificação das lesões, considerando DWNN2, DWNN4 e DWNN6, respectivamente. Neste caso, os resultados foram semelhantes aos do primeiro problema. Os classificadores alcançaram os melhores resultados com o conjunto de dados da DWNN6. Resultados fracos e moderados foram alcançados com DWNN2 e DWNN4. Em geral, tanto o SVM quanto o ELM obtiveram bons resultados. Novamente, com a DWNN6, o SVM-Linear obteve resultados expressivos. SVM-RBF2 e SVM-RBF3 tiveram os piores resultados com cerca de 80% de acurácia e do índice kappa 0,7. O parâmetro gama de 0,50 da RBF corresponde a um limite de decisão gaussiano. Esses resultados para SVM-RBF3 significam que este limite não é adequado para o problema. Contudo, tanto para o problema de detecção de lesão quanto para o problema de classificação de lesões a SVM-RBF2 e SVM-RBF3 obtiveram melhores resultados do que a SVM-RBF1 para os atributos extraídos pela DWNN2 e DWNN4. O que pode indicar que, para esse problema, as fronteiras de decisão criadas pela SVM com o kernel RBF de maior parâmetro γ são mais adequadas para um espaço de menor dimensionalidade. No Apêndice B são dados resultados complementares dos experimentos envolvendo a DWNN com duas, quatro e seis camadas para os problemas de detecção e classificação de lesões.

Figura 54 – Resultados de (a) acurácia e (b) índice kappa dos classificadores com a DWNN2 aplicados para a classificação de lesões.



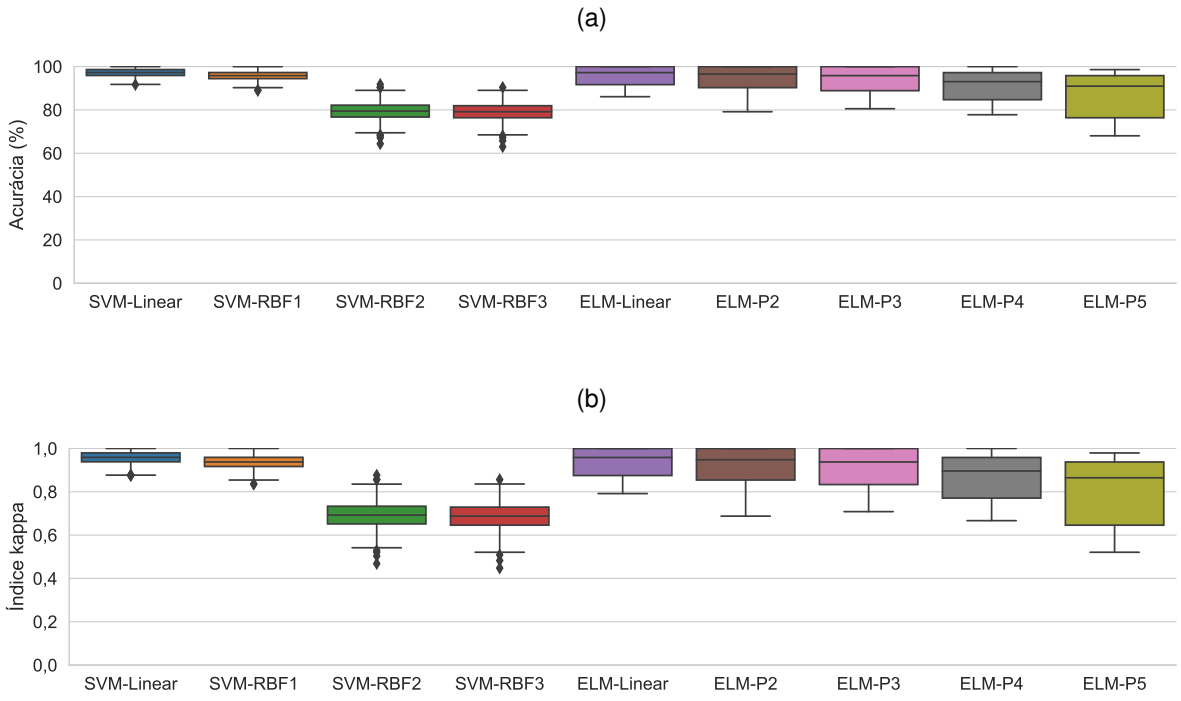
Fonte: O Autor (2022).

Figura 55 – Resultados de (a) acurácia e (b) índice kappa dos classificadores com a DWNN4 aplicados para a classificação de lesões.



Fonte: O Autor (2022).

Figura 56 – Resultados de (a) acurácia e (b) índice kappa dos classificadores com a DWNN6 aplicados para a classificação de lesões.

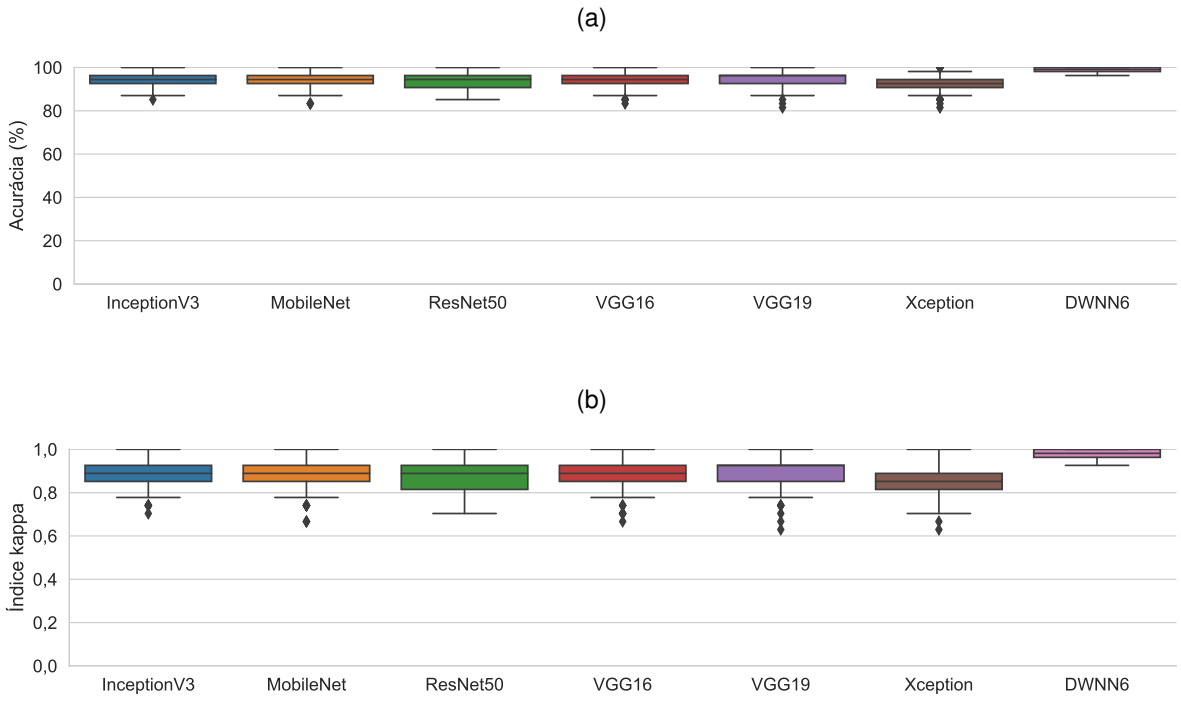


Fonte: O Autor (2022).

Os resultados anteriores mostraram o grande potencial do método DWNN como extrator de atributos. Considerando que os melhores resultados foram obtidos pela DWNN6 com SVM usando *kernel* linear, foram realizados experimentos com seis CNNs do estado da arte pré-treinadas no ImageNet. As CNNs foram aplicadas como extratores de atributos e SVM com *kernel* linear foi utilizada como classificador. As CNNs utilizadas foram: InceptionV3, MobileNet, ResNet50, VGG16, VGG19 e Xception.

As Figuras 57 e 58 mostram *boxplots* de acurácia e índice kappa para as abordagens de detecção e classificação de lesões, respectivamente. Em ambas as abordagens, a SVM-Linear apresentou bons resultados para todos os extratores de atributos. Considerando o problema de detecção de lesões, os resultados das CNNs foram em torno de 95% de acurácia e 0,9 de índice kappa. Entre as seis CNNs o pior resultado foi obtido pela Xception. E os melhores foram obtidos pela InceptionV3 e VGG19, para esta abordagem. Para o problema de classificação de lesões, as CNNs alcançaram entre 80% e 90% de acurácia. Nesse problema os resultados foram mais dispersos, e os melhores resultados das CNNs foram atingidos pela InceptionV3. Em relação à acurácia e ao índice kappa, a DWNN6 superou todos os modelos de CNNs em ambas as abordagens.

Figura 57 – Resultados da SVM-Linear com DWNN6 e seis CNNs do estado da arte aplicadas para o problema de detecção de lesão. Em (a) são mostrados os resultados de acurácia, enquanto que (b) mostra os resultados para o índice kappa.

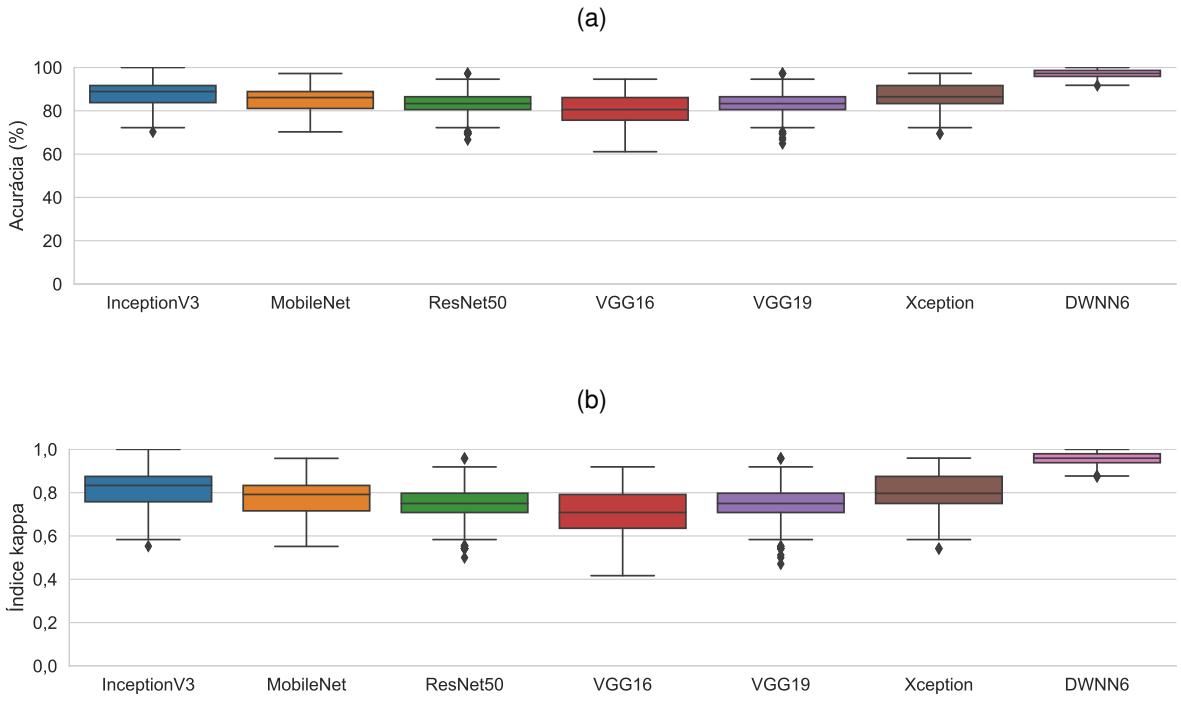


Fonte: O Autor (2022).

As Tabelas 13 e 14 mostram os resultados obtidos pela DWNN6 e pelas CNNs de acordo com a acurácia, índice kappa, sensibilidade, especificidade e precisão para os problemas de detecção e classificação de lesão, respectivamente. Os valores presentes nessas tabelas são a média da amostra e o desvio padrão da amostra de 30 repetições. Os desvios padrão são dados entre parênteses nessas tabelas, e os melhores resultados para cada métrica estão em negrito.

A Tabela 13 mostra os resultados para o problema de detecção de lesão. Analisando esses resultados se pode concluir que a DWNN6 superou as CNNs com relação a todas as métricas avaliadas, obtendo maior média e menor desvio padrão. Assim, com relação a acurácia, índice kappa, sensibilidade, especificidade e precisão, a DWNN6 obteve valores a partir de 4%, 8%, 7%, 1% e 1%, respectivamente, superiores aos das CNNs. Os resultados das CNNs são ligeiramente similares. Mas pode-se afirmar que o melhor resultado entre elas foi obtido pela VGG19, com valores de acurácia, índice kappa, sensibilidade e especificidade superiores ou iguais com relações às outras CNNs.

Figura 58 – Resultados da SVM-Linear com DWNN6 e seis CNNs do estado da arte aplicadas para o problema de classificação de lesão. Em (a) são mostrados os resultados de acurácia, enquanto que (b) mostra os resultados para o índice kappa.



Fonte: O Autor (2022).

Tabela 13 – Desempenho da DWNN6 e CNNs no problema de detecção de lesões de acordo com a acurácia, índice kappa, sensibilidade, especificidade e precisão.

Modelos	Acurácia (%)	Índ. kappa	Sensibilidade	Especificidade	Precisão
DWNN6	**99,0 (0,9)**	**0,98 (0,02)**	**0,98 (0,02)**	**1,00 (0,00)**	**1,00 (0,00)**
InceptionV3	94,9 (3,0)	0,90 (0,06)	0,91 (0,05)	0,98 (0,03)	0,98 (0,03)
MobileNet	93,9 (3,2)	0,88 (0,06)	0,89 (0,06)	0,99 (0,02)	0,99 (0,02)
ResNet50	93,7 (3,3)	0,87 (0,07)	0,89 (0,06)	0,99 (0,02)	0,99 (0,02)
VGG16	94,4 (2,9)	0,89 (0,06)	0,90 (0,06)	0,99 (0,02)	0,99 (0,02)
VGG19	95,0 (3,0)	0,90 (0,06)	0,91 (0,06)	0,99 (0,02)	0,99 (0,02)
Xception	92,8 (3,3)	0,86 (0,07)	0,87 (0,06)	0,98 (0,02)	0,98 (0,02)

Fonte: O Autor (2022).

A Tabela 14 mostra os resultados para a classificação de lesões, que é um problema de classificação com três classes. Desta forma, os valores de sensibilidade, especificidade e precisão apresentados são a média para as três classes. Analisando a tabela se pode verificar que a DWNN6 superou as CNNs no que diz respeito a todas as métricas atingindo os melhores valores e o menor desvio. Nesse sentido, a DWNN6 obteve resultados 10%, 14%, 7%, 5% e 8% superiores aos das CNNs, com relação a acurácia, índice kappa, sensibilidade, especificidade e precisão, respectivamente. Entre as CNNs, os melhores

valores para as métricas foram alcançados por InceptionV3, VGG16 e Xception.

Tabela 14 – Desempenho da DWNN6 e CNNs no problema de classificação de lesões de acordo com a acurácia, índice kappa, sensibilidade, especificidade e precisão.

Modelos	Acurácia (%)	Índ. kappa	Sensibilidade	Especificidade	Precisão
DWNN6	**97,3 (1,9)**	**0,96 (0,03)**	**1,00 (0,01)**	**0,97 (0,02)**	**0,95 (0,04)**
InceptionV3	87,8 (5,2)	0,82 (0,08)	0,91 (0,08)	0,92 (0,05)	0,86 (0,08)
MobileNet	85,1 (5,2)	0,78 (0,08)	0,92 (0,08)	0,90 (0,05)	0,83 (0,08)
ResNet50	83,3 (5,6)	0,75 (0,08)	0,90 (0,09)	0,88 (0,07)	0,80 (0,09)
VGG16	80,8 (6,3)	0,71 (0,09)	0,90 (0,09)	0,87 (0,06)	0,79 (0,09)
VGG19	83,6 (5,7)	0,75 (0,08)	0,90 (0,08)	0,92 (0,05)	0,85 (0,09)
Xception	86,9 (5,5)	0,80 (0,08)	0,93 (0,07)	0,93 (0,05)	0,87 (0,08)

Fonte: O Autor (2022).

Na Tabela 15 são mostrados os tempos de extração de atributos por imagem, em segundos, para cada modelo de extração considerado nesse trabalho. Nessa abordagem, cada método extraiu os atributos de 30 imagens enquanto o tempo de cada extração foi medido. Os resultados apresentados na Tabela 15 são a média e desvio padrão das medições realizadas. O método que entregou os atributos de maneira mais rápida foi a DWNN2. Esse resultado é coerente, considerando que a DWNN2 é um método de arquitetura simples de apenas duas camadas intermediárias. O tempo de execução da DWNN é maior para os modelos com mais camadas intermediárias. Pois com um maior número de camadas intermediárias maior é o número de convoluções realizadas pelo método, e assim, maior é o tempo de execução. Isso pode ser constatado ao analisar o tempo da DWNN4, que é superior ao da DWNN2, e o tempo da DWNN6, superior ao da DWNN4. A DWNN6, arquitetura com os melhores resultados da DWNN, obteve seus resultados em 7,62 segundos em média. Entre os resultados das CNNs, apenas as VGG16 e VGG19 obteve resultados com tempo menores do que a DWNN6, ambos por volta dos 4 segundos. O tempo da DWNN6 é equiparado ao tempo da MobileNet, 8,05 s, e duas vezes menor do que o tempo da ResNet50, com 16,1 s. Os modelos que mais demoraram a extrair os atributos foram a Inception V3 com 7,62 s e Xception com 34 s. A partir dos resultados mostrados nas Tabelas 14 e 13 foi possível analisar que a DWNN é capaz de entregar resultados satisfatórios de classificação comparáveis e até superiores aos resultados de CNNs do estado da arte. Por outro lado, ao analisar a Tabela 15 verifica-se que o tempo de execução da DWNN é superior apenas às VGGs, contudo a DWNN6 obteve melhores resultados ou consideravelmente equiparáveis aos das VGGs. Comparado com os tempos

obtidos pelas Inception V3 e Xception, o tempo da DWNN6 foi de até 4,5 vezes menor. E ainda assim os resultados da DWNN6 foram melhores do que essas CNNs para os problemas de detecção e classificação de lesões.

Tabela 15 – Tempo de extração de atributos por imagem em segundos.

Modelo	Tempo por imagem (s)
DWNN2	$2{,}21 \pm 0{,}01$
DWNN4	$4{,}92 \pm 0{,}02$
DWNN6	$7{,}62 \pm 0{,}03$
Inception V3	$21{,}3 \pm 0{,}7$
MobileNet	$8{,}05 \pm 0{,}09$
ResNet50	$16{,}1 \pm 0{,}4$
VGG16	$4{,}35 \pm 0{,}07$
VGG19	$4{,}46 \pm 0{,}06$
Xception	34 ± 2

Fonte: O Autor(2022).

As Figuras 59 e 60 mostram as matrizes de confusão, da DWNN6 e das CNNs, para os problemas de detecção e classificações de lesões, respectivamente. Essas matrizes foram obtidas ao classificar os dados obtidos pelos extratores de atributos com a SVM-Linear. De uma modo geral, como é possível observar nas Tabelas 13 e 14, os resultados são satisfatórios, mas a partir das matrizes de confusão é possível analisar as classes mais difíceis de classificar, a partir da análise das classes que tiveram mais instâncias classificadas de maneira errada.

Para o problema de detecção de lesões, se observa que a SVM-Linear apresentou falsos negativos em todos os experimentos considerando a DWNN6 e as CNNs. Ou seja, a classe que apresentou mais confusão entre os experimentos foi a classe com lesão. Para a DWNN6 o erro foi de apenas 5 instâncias, já para a Xception, o pior caso, o erro foi de 36 instâncias. Para a classificação da classe sem lesão a DWNN6 não apresentou falhas. Contudo todas as CNNs apresentaram algum erro para essa classe. Por exemplo, o erro com a MobileNet foi de apenas 1 instância, enquanto que, para a Inception V3, o erro foi de 7 instâncias, sendo esse o pior caso entre as CNNs. Dessa maneira, as imagens que possuem alguma lesão mamária apresentaram mais dificuldade para os algoritmos do que a classe com lesão.

Para o problema de classificação de lesões, a classe que apresentou mais confusão foi a classe lesão benigna, sendo esta, a classe que os métodos tiveram mais dificuldade em

Figura 59 – Matrizes de confusão para o problema de detecção de lesões.

DWNN6

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	265	5
Sem Lesão	0	270

Inception V3

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	246	24
Sem Lesão	7	263

MobileNet

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	233	37
Sem Lesão	1	269

ResNet50

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	235	35
Sem Lesão	6	264

VGG16

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	246	24
Sem Lesão	2	268

VGG19

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	245	25
Sem Lesão	2	268

Xception

	Classificado como	
	Com Lesão	Sem Lesão
Com Lesão	234	36
Sem Lesão	4	266

Fonte: O Autor (2022).

classificar. Para a DWNN6, a SVM-Linear avaliou erroneamente 7 instâncias, considerando 5 delas como pertencentes à classe de cisto, e duas delas à classe de lesão maligna. Entre as CNNs, no melhor cenário está a Inception V3, com 20 instâncias mal classificadas, e o pior caso foi obtido com a VGG16, com 37 instâncias da classe lesão benigna classificadas de forma errada. Entre as classes de cisto e lesão maligna não houve unanimidade entre os métodos. Por exemplo, para a DWNN6, o classificador errou apenas 1 instância da classe cisto, considerando-a como pertencente à classe lesão benigna, mas não errou nenhuma instância para a classe de lesão maligna. Por outro lado, os resultados com a Inception V3 e ResNet50 apresentaram os mesmo número de instâncias erradas para cisto e lesão maligna. Sendo 9 instâncias classificadas de forma errada com a Inception e 13 com a

ResNet50, para ambas as classes. Já os resultados com MobileNet, VGG16, VGG19 e Xception os erros com as instâncias de lesão maligna foram superiores aos erros da classe cisto.

Figura 60 – Matrizes de confusão para o problema de classificação de lesões.

DWNN6

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	120	1	0
L. Benigna	5	114	4
L. Maligna	0	0	121

Inception V3

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	112	9	0
L. Benigna	12	101	8
L. Maligna	4	7	110

MobileNet

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	112	8	1
L. Benigna	17	92	12
L. Maligna	9	9	103

ResNet50

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	108	12	1
L. Benigna	22	85	14
L. Maligna	2	11	108

VGG16

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	109	8	4
L. Benigna	22	84	15
L. Maligna	9	10	102

VGG19

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	110	9	2
L. Benigna	12	93	16
L. Maligna	3	12	106

Xception

	Classificada como		
	Cisto	Lesão Benigna	Lesão Maligna
Cisto	113	5	3
L. Benigna	12	94	15
L. Maligna	2	11	108

Fonte: O Autor (2022).

Como pode ser visto na Tabela 9, a DWNN é um método de extração de atributos simples e sem parâmetros de treinamento. Pois na DWNN não é preciso ajustar os pesos

nas camadas intermediárias, e sim definir um banco de filtros lineares e a função de *pooling*. Por outro lado, as CNNs possuem uma grande quantidade de parâmetros treináveis, e por isso, precisam de um grande conjunto de dados para os seus treinamentos. Assim, para utilizar as CNNs para problemas de conjuntos de dados pequenos, como o problema desse trabalho, muitas vezes se faz necessário utilizar estratégias da transferência de aprendizagem. Uma dessas estratégias é utilizar as redes profundas pré-treinadas como extratores de atributos, tendo sido treinadas em conjuntos de treino diferentes do problema de interesse, i.e. classificação de imagens de termografia de mama.

As principais vantagens de usar transferência de aprendizagem com redes profundas neste problema são: a capacidade de extração de atributos implícitos, sem a necessidade de conhecimento especialista humano; incorporação de complexidade de representação, devido à utilização de um número grande de camadas, o que torna a representação vetorial da saída adequada para fronteiras de decisão complexas. No entanto, essa complexidade também leva a uma representação esparsa, sendo necessário utilizar classificadores específicos para esses problemas. No entanto, o problema da dimensionalidade pode ser resolvido com a utilização de algoritmos de seleção de atributos, como a *Random Forest*, capaz de selecionar os atributos estatisticamente mais relevantes. Isso teve como consequência a obtenção de bons resultados de classificação para máquinas de vetor de suporte com kernel linear. Os resultados experimentais mostram, portanto, que esta pode ser uma boa estratégia para a adoção da transferência de aprendizagem com redes profundas em problemas de classificação de imagens biomédicas.

Com relação à DWNN, verifica-se que com o aumento do número de camadas, há um aumento do número de atributos extraídos por imagem, resultando em um melhor desempenho para os classificadores. Isso ficou constatados com os resultados intermediários para com 2 e 4 camadas intermediárias. Resultados bastante expressivos foram obtidos com 6 camadas intermediárias. Os resultados com a DWNN6 foram superiores àqueles obtidos com redes profundas do estado da arte, com acurácia, índice kappa, sensibilidade, especificidade e precisão superiores a 95%. Com relação ao tempo de extração de atributos, entre as CNNs, apenas as VGG16 e VGG19 obtiveram tempos menores do que a DWNN6. Do ponto de vista biomédico, esses resultados mostram que a DWNN hibridizada com seleção de atributos por *Random Forest* e classificação final com SVM com kernel linear são uma abordagem que pode otimizar o uso de termografia de mama na prática clínica como

processo de triagem para técnicas mais sofisticadas e de maior custo, como a mamografia. A otimização da tomada de decisão clínica a partir da análise de imagens termográficas por aprendizado de máquina também pode contribuir para uma maior difusão do uso clínico desta técnica mesmo ainda não existindo um amplo conjunto de profissionais que possam extrair informações da análise visual de imagens térmicas de mama.

6 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

O câncer de mama é o tipo de câncer mais frequente e mortal entre as mulheres. Contudo, as chances de recuperação de uma paciente com câncer de mama aumentam se a doença for detectada em seus estágios iniciais. A termografia de mama é uma técnica de imagem que foi aprovada em 1982 pela FDA como técnica auxiliar à mamografia no rastreamento do câncer de mama. Contudo, a correta interpretação das imagens de infravermelho ainda continua sendo um desafio. Este trabalho propôs um sistema inteligente, baseado em técnicas da aprendizagem profunda, para a detecção e classificação de lesões de mama.

Além disso, nesse trabalho, foi proposto um método da aprendizagem profunda baseada na Transformada de *wavelets* para classificação de imagens, a *Deep-Wavelet Neural Network* (DWNN). Na DWNN, não é preciso ajustar os pesos, e sim definir um banco de filtros lineares e a função de *pooling*. Essa técnica foi aplicada ao problema de detecção e classificação de lesões de mama em imagens térmicas. A DWNN foi comparada a um conjunto de redes profundas baseadas em *Convolutional Neural Networks* (CNNs) do estado da arte. Para as CNNs a estratégia utilizada foi a da transferência de aprendizagem, onde as redes profundas pré-treinadas são utilizadas como etapas de extração de atributos, tendo sido treinadas em conjuntos de treino diferentes do problema de interesse, i.e. classificação de imagens de termografia de mama.

Os experimentos foram realizados de acordo com as abordagens de detecção e classificação de lesões. A primeira abordagem, um problema binário, envolve as imagens com e sem lesão mamária. Por outro lado, a classificação de lesões é um problema multi-classe realizado com as imagens de pacientes com cisto, lesão benigna e maligna. Nesse contexto, as técnicas da aprendizagem profunda desempenharam resultados satisfatórios para a classificação de imagens de termografia de mama. Os resultados da DWNN e das CNNs foram superiores a 80% para ambas abordagens considerandos as métricas acurácia, índice kappa, sensibilidade, especificidade e precisão.

Com relação à DWNN, a melhor arquitetura, para este problema, foi a de seis camadas com *pooling* de máximo, com SVM linear como classificador na camada de saída e tendo seus atributos selecionados através da *random forest*. Os resultados com esta

arquitetura foram superiores àqueles obtidos com redes profundas do estado da arte no contexto da classificação de lesão considerando todas as métricas consideradas nesse trabalho.

Desta forma, comparada com as CNNs do estado da arte, a DWNN obteve resultados competitivos e até superiores. Apesar disso, a DWNN é uma técnica mais simples em termos de arquitetura o qual não necessita de treinamento. À vista disso, não necessita de um banco de imagens com um número elevado de exemplos como as redes convolucionais. Fator esse que pode ser crucial no contexto de aplicações médicas, como a termografia de mama, onde a aquisição de novas imagens é um processo custoso, trabalhoso e que exige a avaliações de profissionais da saúde especializados. No que diz respeito ao tempo de execução, a DWNN entrega seus resultados em um tempo um pouco maior com relação as VGG16 e VGG19. Mas um tempo equiparável ao da MobileNet, e consideravelmente menor com relação aos tempos das ResNet50, Inception V3 e Xception.

Para trabalhos futuros, pode-se sugerir:

- O desenvolvimento de um sistema web ou mobile, baseado em ferramentas inteligentes, para a análise de imagens de termografia para o apoio ao diagnóstico do câncer de mama. O objetivo dessa ferramenta é servir como um instrumento de triagem para a detecção de lesões mamária. Especialmente em regiões de poucos recursos onde o acesso à mamografia é difícil, e onde quantidade de profissionais especialistas disponíveis é baixa.

- A realização de experimentos com a DWNN com uma base de dados maior e menos desbalanceada. A ampliação da base de dados, poderá ser feita através da captação de novas imagens, ou através de parcerias com outras instituições que trabalhem com termografia de mama.

- O método da DWNN é sequencial, isto é uma camada é chamada apenas quando o processamento da camada anterior é finalizado. Como trabalho futuro, pode-se sugerir retirar essa limitação sequencial do método da DWNN para que seja possível a implementação de estratégias de paralelismo para executar o algoritmo com a finalidade de diminuir o tempo de avaliação das imagens de entrada.

REFERÊNCIAS

ABE, S. Analysis of multiclass support vector machines. *Thyroid*, v. 21, n. 3, p. 3772, 2003.

AFFONSO, E. T. F.; SILVA, A. M.; SILVA, M. P.; RODRIGUES, T. M. D.; MOITA, G. F. Uso redes neurais multilayer perceptron (MLP) em sistema de bloqueio de websites baseado em conteúdo. *Volume XXIX. Number 93. Computational Intelligence Techniques for Optimization and Data Modeling*, American Psychological Association, p. 9075–9090, 2010.

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. *2017 International Conference on Engineering and Technology (ICET)*. [S.l.], 2017. p. 1–6.

ARAÚJO, M. C.; LIMA, R. C.; SOUZA, R. M. D. Interval symbolic feature extraction for thermography breast cancer detection. *Expert Systems with Applications*, Elsevier, v. 41, n. 15, p. 6728–6737, 2014.

ARAÚJO, M. C. d. Utilização de câmera por infravermelho para avaliação de diferentes patologias em clima tropical e uso conjunto de sistemas de banco de dados para detecção de câncer de mama [dissertation]. *Recife, PE, Brazil: Federal University of Pernambuco*, 2009.

ARORA, N.; MARTINS, D.; RUGGERIO, D.; TOUSIMIS, E.; SWISTEL, A. J.; OSBORNE, M. P.; SIMMONS, R. M. Effectiveness of a noninvasive digital infrared thermal imaging system in the detection of breast cancer. *The American Journal of Surgery*, Elsevier, v. 196, n. 4, p. 523–526, 2008.

AZEVEDO, W. W.; LIMA, S. M.; FERNANDES, I. M.; ROCHA, A. D.; CORDEIRO, F. R.; Da Silva-Filho, A. G.; Dos Santos, W. P. Fuzzy Morphological Extreme Learning Machines to detect and classify masses in mammograms. *IEEE International Conference on Fuzzy Systems*, v. 2015-Novem, 2015. ISSN 10987584.

BAFFA, M. de F. O.; LATTARI, L. G. Convolutional neural networks for static and dynamic breast infrared imaging classification. In: *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2018. p. 174–181.

BARTLETT, P. L. For valid generalization the size of the weights is more important than the size of the network. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1997. p. 134–140.

BERRAR, D. *Cross-Validation.* 2019.

BEZERRA, E. Introdução à aprendizagem profunda. *Artigo–31º Simpósio Brasileiro de Banco de Dados–SBBD2016–Salvador*, 2016.

BEZERRA, L.; OLIVEIRA, M.; ROLIM, T.; CONCI, A.; SANTOS, F.; LYRA, P.; LIMA, R. C. Estimation of breast tumor thermal properties using infrared images. *Signal Processing*, Elsevier, v. 93, n. 10, p. 2851–2863, 2013.

BEZERRA, L. A. *Uso de imagens termográficas em tumores mamários para validação de simulação computacional*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2007.

BLAGUS, R.; LUSA, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, Springer, v. 14, n. 1, p. 106, 2013.

BORCHARTT, T. B.; CONCI, A.; LIMA, R. C.; RESMINI, R.; SANCHEZ, A. Breast thermography from an image processing viewpoint: A survey. *Signal Processing*, Elsevier, v. 93, n. 10, p. 2785–2803, 2013.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the fifth annual workshop on Computational learning theory*. [S.l.: s.n.], 1992. p. 144–152.

BOUCKAERT, R. R.; FRANK, E.; HALL, M.; KIRKBY, R.; REUTEMANN, P.; SEEWALD, A.; SCUSE, D. Weka manual for version 3-8-1. *The University of Waikato: Hamilton, New Zealand*, 2016.

BRAGA, A. d. P.; CARVALHO, A. P. d. L. F. d.; LUDERMIR, T. B. Fundamentos de redes neurais artificiais. *Rio de Janeiro: 11a Escola de Computação*, 1998.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001.

BURGER, W.; BURGE, M. J. *Digital image processing: an algorithmic introduction using Java*. [S.l.]: Springer, 2016.

CAMBRIA, E.; HUANG, G.-B.; KASUN, L. L. C.; ZHOU, H.; VONG, C. M.; LIN, J.; YIN, J.; CAI, Z.; LIU, Q.; LI, K. et al. Extreme learning machines [trends & controversies]. *IEEE intelligent systems*, IEEE, v. 28, n. 6, p. 30–59, 2013.

CASTRO, P. A. D. de; SANTORO, D. M.; CAMARGO, H. A.; NICOLETTI, M. C. Improving a pittsburgh learnt fuzzy rule base using feature subset selection. In: IEEE. *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*. [S.l.], 2004. p. 180–185.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, Elsevier, v. 40, n. 1, p. 16–28, 2014.

CHAVES, E.; GONÇALVES, C. B.; ALBERTINI, M. K.; LEE, S.; JEON, G.; FERNANDES, H. C. Evaluation of transfer learning of pre-trained CNNs applied to breast cancer detection on infrared images. *Applied Optics*, Optical Society of America, v. 59, n. 17, p. E23–E28, 2020.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, v. 16, p. 321–357, 2002.

CHEN, B.; CHEN, L.; CHEN, Y. Efficient ant colony optimization for image feature selection. *Signal processing*, Elsevier, v. 93, n. 6, p. 1566–1576, 2013.

CHEREPENIN, V.; KARPOV, A.; KORJENEVSKY, A.; KORNIENKO, V.; MAZALETSKAYA, A.; MAZOUROV, D.; MEISTER, D. A 3D electrical impedance tomography (EIT) system for breast cancer detection. *Physiological measurement*, IOP Publishing, v. 22, n. 1, p. 9, 2001.

CHOLLET, F. *Keras*. [S.l.]: GitHub, 2015. <https://github.com/fchollet/keras>.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1251–1258.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

CUTLER, A.; CUTLER, D. R.; STEVENS, J. R. Random forests. In: *Ensemble Machine Learning - Methods and Applications*. [S.l.]: Springer, 2012. p. 157–175.

CYBENKO, G. *Continuous valued neural networks with two hidden layers are sufficient*. [S.l.], 1988.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, Springer, v. 2, n. 4, p. 303–314, 1989.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

EKICI, S.; JAWZAL, H. Breast cancer diagnosis using thermography and convolutional neural networks. *Medical Hypotheses*, v. 137, p. 109542, 2020. ISSN 0306-9877.

ESPINDOLA, N. A. *Estimativa de parâmetros termofísicos da mama e de suas anomalias a partir do mapeamento de temperaturas da superfície de imagens por infravermelho*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017.

ETEHADTAVAKOL, M.; NG, E. Y. Breast thermography as a potential non-contact method in the early detection of cancer: a review. *Journal of Mechanics in Medicine and Biology*, World Scientific, 2013.

EULER-CHELPIN, M. von; LILLHOLM, M.; VEJBORG, I.; NIELSEN, M.; LYNGE, E. Sensitivity of screening mammography by density and texture: a cohort study from a population-based screening program in denmark. *Breast Cancer Research*, BioMed Central, v. 21, n. 1, p. 1–7, 2019.

GOGOI, U. R.; BHOWMIK, M. K.; BHATTACHARJEE, D.; GHOSH, A. K. Singular value based characterization and analysis of thermal patches for early breast abnormality detection. *Australasian physical & engineering sciences in medicine*, Springer, v. 41, n. 4, p. 861–879, 2018.

GOMES, J. C.; BARBOSA, V. A. d. F.; SANTANA, M. A.; BANDEIRA, J.; VALENÇA, M. J. S.; SOUZA, R. E. de; ISMAEL, A. M.; SANTOS, W. P. dos. Ikonos: An intelligent tool to support diagnosis of covid-19 by texture analysis of x-ray images. *Research on Biomedical Engineering*, Springer, p. 1–14, 2020.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, v. 11, n. 1, p. 10–18, 2009.

HALL, M. A. Correlation-based feature selection for machine learning. University of Waikato Hamilton, 1999.

HALLIDAY; RESNICK; WALKER. *Fundamentos de Física - Gravitação, Ondas e Termodinâmica - Volume 2*. 10. ed. [S.l.]: LTC, 2016.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. H. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, IEEE, n. 6, p. 610–621, 1973.

HASAN, M. A. M.; NASSER, M.; AHMAD, S.; MOLLA, K. I. Feature selection for intrusion detection using random forest. *Journal of information security*, Scientific Research Publishing, v. 7, n. 3, p. 129–140, 2016.

HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman, 2001. ISBN 9788573077186.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

HEAD, J. F.; LIPARI, C. A.; WANG, F.; DAVIDSON, J. E.; ELLIOTT, R. Application of second generation infrared imaging with computerized image analysis to breast cancer risk assessment. In: IEEE. *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.], 1996. v. 5, p. 2093–2094.

HERBRICH, R. *Learning kernel classifiers: theory and algorithms*. [S.l.]: MIT press, 2001.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, 2017.

HUANG, G.-B.; SIEW, C.-K. Extreme learning machine with randomly assigned RBF kernels. *International Journal of Information Technology*, v. 11, n. 1, p. 16–24, 2005.

HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE. *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*. [S.l.], 2004. v. 2, p. 985–990.

HUANG, G. B.; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: Theory and applications. *Neurocomputing*, v. 70, n. 1-3, p. 489–501, 2006. ISSN 09252312.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. *Diretrizes para a Detecção Precoce do Câncer de mama no Brasil*. Rio de Janeiro, 2015.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. *Câncer de mama*. [S.l.], 2018. Acesso em: 16 jan. 2019. Disponível em: <https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama>.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. *ESTIMATIVA 2018| Incidência de câncer no Brasil*. Rio de Janeiro, 2018. Acesso em: 22 jan. 2019. Disponível em: <http://www1.inca.gov.br/estimativa/2018/>.

INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. *Estatísticas de câncer*. [S.l.], 2021. Acesso em: 14 nov. 2021. Disponível em: <https://www.inca.gov.br/numeros-de-cancer>.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 448–456.

JOHANSSON, E. M.; DOWLA, F. U.; GOODMAN, D. M. Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, World Scientific, v. 2, n. 04, p. 291–301, 1991.

JOVIĆ, A.; BRKIĆ, K.; BOGUNOVIĆ, N. A review of feature selection methods with applications. In: IEEE. *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. [S.l.], 2015. p. 1200–1205.

KAN, C.; SRINATH, M. D. Combined features of cubic b-spline wavelet moments and zernike moments for invariant character recognition. In: IEEE. *Proceedings International Conference on Information Technology: Coding and Computing*. [S.l.], 2001. p. 511–515.

KARIM, C. N.; MOHAMED, O.; RYAD, T. A new approach for breast abnormality detection based on thermography. *Medical Technologies Journal*, v. 2, n. 3, p. 245–254, 2018.

KARPATHY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; FEI-FEI, L. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1725–1732.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, p. 1097–1105, 2012.

LAWSON, R. Implications of surface temperatures in the diagnosis of breast cancer. *Canadian Medical Association Journal*, Canadian Medical Association, v. 75, n. 4, p. 309, 1956.

LIANG, H.; LIN, X.; ZHANG, Q.; KANG, X. Recognition of spoofed voice using convolutional neural networks. In: IEEE. *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. [S.l.], 2017. p. 293–297.

LIMA, A. M. *A maldição da dimensionalidade*. 2020. Acesso em: 11 fev. 2021. Disponível em: <https://eaulas.usp.br/portal/video.action?idItem=9667>.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007.

MALLAT, S. G. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoustics, Speech, and Signal Processing*, v. 37, n. 12, p. 2091–2110, 1989.

MAMBOU, S. J.; MARESOVA, P.; KREJCAR, O.; SELAMAT, A.; KUCA, K. Breast cancer detection using infrared thermal imaging and a deep learning model. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 9, p. 2799, 2018.

MATHWORKS. *2-D Discrete Wavelet Analysis*. 2021. Acesso em: 20 fev. 2021. Disponível em: <https://www.mathworks.com/help/wavelet/ug/two-dimensional-discrete-wavelet-analysis.html>.

MEIRA, L. F.; KRUEGER, E.; NEVES, E. B.; NOHAMA, P.; SOUZA, M. A. de. Termografia na área biomédica. *Pan American Journal of Medical Thermology*, p. 31–41, 2014.

MELO, J. R. F. d. et al. *Metodologia para desenvolvimento de geometria tridimensional de mama e seu uso na estimativa de parâmetros termofísicos usando imagens termográficas*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.

MICHE, Y.; SORJAMAA, A.; BAS, P.; SIMULA, O.; JUTTEN, C.; LENDASSE, A. OP-ELM: optimally pruned extreme learning machine. *IEEE transactions on neural networks*, IEEE, v. 21, n. 1, p. 158–162, 2009.

MISHRA, S.; PRAKASH, A.; ROY, S. K.; SHARAN, P.; MATHUR, N. Breast cancer detection using thermal images and deep learning. In: IEEE. *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*. [S.l.], 2020. p. 211–216.

MOGHBEL, M.; MASHOHOR, S. A review of computer assisted detection/diagnosis (CAD) in breast thermography for breast cancer detection. *Artificial Intelligence Review*, Springer, 2013.

MORALES-CERVANTES, A.; KOLOSOVAS-MACHUCA, E. S.; GUEVARA, E.; REDUCINDO, M. M.; HERNÁNDEZ, A. B. B.; GARCÍA, M. R.; GONZÁLEZ, F. J. An automated method for the evaluation of breast cancer using infrared thermography. *EXCLI journal*, Leibniz Research Centre for Working Environment and Human Factors, v. 17, p. 989, 2018.

MULLER, K.-R.; MIKA, S.; RATSCH, G.; TSUDA, K.; SCHOLKOPF, B. An introduction to kernel-based learning algorithms. *IEEE transactions on neural networks*, IEEE, v. 12, n. 2, p. 181–201, 2001.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *ICML*. [S.l.: s.n.], 2010.

NEUMANN, Ł.; NOWAK, R. M.; OKUNIEWSKI, R.; OLESZKIEWICZ, W.; CICHOSZ, P.; JAGODZIŃSKI, D.; MATYSIEWICZ, M. Preprocessing for classification of thermograms in breast cancer detection. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*. [S.l.], 2016. v. 10031, p. 100313A.

OLESZKIEWICZ, W.; CICHOSZ, P.; JAGODZIŃSKI, D.; MATYSIEWICZ, M.; NEUMANN, Ł.; NOWAK, R. M.; OKUNIEWSKI, R. Application of SVM classifier in thermographic image classification for early detection of breast cancer. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016*. [S.l.], 2016. v. 10031, p. 100312T.

OLIVEIRA, M. M. d. Desenvolvimento de protocolo e construção de um aparato mecânico para padronização da aquisição de imagens termográficas de mama [dissertation]. *Recife: Federal University of Pernambuco*, 2012.

ORGANIZAÇÃO MUNDIAL DA SAÚDE. *Breast cancer*. [S.l.], 2019. Acesso em: 16 jan. 2019. Disponível em: <https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/>.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 22, n. 10, p. 1345–1359, 2009.

QI, Y. Random forest for bioinformatics. In: *Ensemble machine learning*. [S.l.]: Springer, 2012. p. 307–323.

QUEIROZ, K. F. F. d. C. *Desenvolvimento e implementação de uma ferramenta computacional de uso médico para análise de imagens termográficas*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2016.

REFAEILZADEH, P.; TANG, L.; LIU, H. Cross-validation. In: ____. *Encyclopedia of Database Systems*. New York, NY: Springer New York, 2016. p. 1–7. ISBN 978-1-4899-7993-3. Disponível em: <https://doi.org/10.1007/978-1-4899-7993-3_565-2>.

RIEDMILLER, M. Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, Elsevier, v. 16, n. 3, p. 265–278, 1994.

RODRIGUES, A. L.; SANTANA, M. A. de; AZEVEDO, W. W.; BEZERRA, R. S.; BARBOSA, V. A.; LIMA, R. C. de; SANTOS, W. P. dos. Identification of mammary lesions in thermographic images: feature selection study using genetic algorithms and particle swarm optimization. *Research on Biomedical Engineering*, Springer, v. 35, n. 3, p. 213–222, 2019.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.

ROSEBROCK, A. *Transfer Learning with Keras and Deep Learning*. [S.l.], 2019. Acesso em: 09 mar. 2021. Disponível em: <https://www.pyimagesearch.com/2019/05/20/transfer-learning-with-keras-and-deep-learning/>.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

ROSLIDAR, R.; SADDAMI, K.; ARNIA, F.; SYUKRI, M.; MUNADI, K. A study of fine-tuning CNN models based on thermal imaging for breast cancer classification. In: IEEE. *2019 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. [S.l.], 2019. p. 77–81.

SÁNCHEZ-RUIZ, D.; OLMOS-PINEDA, I.; OLVERA-LÓPEZ, J. A. Automatic region of interest segmentation for breast thermogram image classification. *Pattern Recognition Letters*, Elsevier, v. 135, p. 72–81, 2020.

SANDRI, M.; ZUCCOLOTTO, P. Variable selection using random forests. In: *Data analysis, classification and the forward search*. [S.l.]: Springer, 2006. p. 263–270.

SANTANA, M. A. d. *Sistemas inteligentes para apoio ao diagnóstico do câncer de mama usando imagens mamográficas e termográficas*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2020.

SANTANA, M. A. d.; PEREIRA, J. M. S.; SILVA, F. L. d.; LIMA, N. M. d.; SOUSA, F. N. d.; ARRUDA, G. M. S. d.; LIMA, R. d. C. F. d.; SILVA, W. W. A. d.; SANTOS, W. P. d. Breast cancer diagnosis based on mammary thermography and extreme learning machines. *Research on Biomedical Engineering*, 2018.

SARKAR, D. *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*. 2018. Acesso em: 05 mar. 2021. Disponível em: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>.

SCHAEFER, G.; NAKASHIMA, T.; ZAVISEK, M. Analysis of breast thermograms based on statistical image features and hybrid fuzzy classification. In: SPRINGER. *International Symposium on Visual Computing*. [S.l.], 2008. p. 753–762.

SHIN, H.-C.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; YAO, J.; MOLLURA, D.; SUMMERS, R. M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, IEEE, v. 35, n. 5, p. 1285–1298, 2016.

SILVA, A. L. R. d. *Seleção de atributos para apoio ao diagnóstico do câncer de mama usando imagens termográficas, algoritmos genéticos e otimização por enxame de partículas*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.

SILVA, A. S. V. d. *Classificação e segmentação de termogramas de mama para triagem de pacientes residentes em regiões de poucos recursos médicos*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2015.

SILVA, L.; SAADE, D.; SEQUEIROS, G.; SILVA, A.; PAIVA, A.; BRAVO, R.; CONCI, A. A new database for breast research with infrared image. *Journal of Medical Imaging and Health Informatics*, American Scientific Publishers, v. 4, n. 1, p. 92–100, 2014.

SILVA, L.; SEIXAS, F.; FONTES, C.; MUCHALUAT-SAADE, D.; CONCI, A. A computational method for breast abnormality detection using thermographs. In: IEEE. *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*. [S.l.], 2020. p. 469–474.

SILVA, L. F.; SANTOS, A. A. S.; BRAVO, R. S.; SILVA, A. C.; MUCHALUAT-SAADE, D. C.; CONCI, A. Hybrid analysis for indicating patients with breast cancer using temperature time series. *Computer methods and programs in biomedicine*, Elsevier, v. 130, p. 142–153, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

SINGH, D.; SINGH, A. K. Role of image thermography in early breast cancer detection- past, present and future. *Computer Methods and Programs in Biomedicine*, v. 183, p. 105074, 2020. ISSN 0169-2607.

SKANSI, S. *Introduction to Deep Learning: from logical calculus to artificial intelligence*. [S.l.]: Springer, 2018.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2017. v. 31, n. 1.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826.

TABAA, M.; FAHMANI, H.; BENSAG, H. et al. Covid-19's rapid diagnosis open platform based on x-ray imaging and deep learning. *Procedia Computer Science*, Elsevier, v. 177, p. 618–623, 2020.

TELLO-MIJARES, S.; WOO, F.; FLORES, F. Breast cancer identification via thermography image segmentation with a gradient vector flow and a convolutional neural network. *Journal of healthcare engineering*, Hindawi, v. 2019, 2019.

TRAN, D.; WANG, H.; TORRESANI, L.; FEISZLI, M. Video classification with channel-separated convolutional networks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 5552–5561.

VASCONCELOS, J. H. d. *Investigações sobre métodos de classificação para uso em termografia de mama*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017.

VERLEYSEN, M.; FRANÇOIS, D. The curse of dimensionality in data mining and time series prediction. In: *Computational Intelligence and Bioinspired Systems*. [S.l.]: Springer Berlin Heidelberg, 2005. p. 758–770. ISBN 978-3-540-32106-4.

WALKER, D.; KACZOR, T. Breast thermography: history, theory, and use. Is this screening tool adequate for standalone use? *Natural Medicine Journal*, v. 4, n. 7, 2012.

WILLIAMS, R.; WILLIAMS, G. Pioneers of invisible radiation photography. *Medical and Scientific Photography*, 2002.

YANG, F.; SUN, Q.; JIN, H.; ZHOU, Z. Superpixel segmentation with fully convolutional networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 13964–13973.

ZHOU, R.; LIU, F.; GRAVELLE, C. W. Deep learning for modulation recognition: a survey with a demonstration. *IEEE Access*, IEEE, v. 8, p. 67366–67376, 2020.

APÊNDICE A – CONTRIBUIÇÕES PARA O TEMA

As principais contribuições deste trabalho foram:

- A formalização matemática da *Deep-Wavelet Neural Network*;
- A aplicação da *Deep-Wavelet Neural Network* e redes neurais convolucionais para a extração de atributos de imagens de termografia;
- Balanceamento das bases, no mapeamento das características, através do algoritmo SMOTE;
- A aplicação da *Random Forest* para a seleção dos atributos mais relevantes em bases obtidas pelos extratores de atributos baseados na aprendizagem profunda;
- O desenvolvimento do sistema de detecção e classificação de lesões de mama.

A Tabela A1 mostra os trabalhos publicados durante a execução desta pesquisa. Ao total foram um artigo em evento nacional, dois artigos de revista científica e um capítulo de livro, esses últimos sendo internacionais. O evento nacional da publicação foi o Congresso Brasileiro de Automática (CBA). Já os periódicos foram publicados na *Medical & Biological Engineering & Computing*, e na *Research on Biomedical Engineering*. O capítulo de livro foi publicado no livro *Deep Learning for Data Analytics: Foundations, Biomedical Applications, and Challenges*. Além disso, pelo trabalho com a *Deep-Wavelet Neural Network* na classificação de imagens de termografia, este autor recebeu, juntamente com Maíra Araújo de Santana e Jessiane Mônica Silva Pereira, a menção honrosa por ter ficado em quarto lugar no Desafio de Aplicações Bio-Tech (BMEBioBrazil 2019).

Tabela A1 – Trabalhos publicados durante este trabalho

Evento/Periódico/Livro	Título do artigo/capítulo	Ano
Congresso Brasileiro de Automática (CBA) - João Pessoa	Desempenho de máquinas de aprendizado extremo com operadores morfológicos para identificação e classificação de lesões em imagens frontais de termografia de mama.	2018
Deep Learning for Data Analytics: Foundations, Biomedical Applications, and Challenges	Deep-Wavelet Neural Networks for breast cancer early diagnosis using mammary termographies	2020
Research on Biomedical Engineering	Identification of mammary lesions inthermographic images: feature selection study using genetic algorithms and particle swarm optimization	2019
Research on Biomedical Engineering	Feature selection based on dialectics to support breast cancer diagnosis using thermographic images	2021

Fonte: O Autor (2022).

APÊNDICE B – RESULTADOS COMPLEMENTARES DA DWNN

Nas Tabelas B1 e B2 são dados os valores de acurácia, índice kappa, sensibilidade, especificidade e precisão para a DWNN com duas, quatro e seis camadas considerando todos os classificadores utilizado nesse trabalho, para os problemas de detecção e classificação de lesões, respectivamente. Os valores presentes na tabela são a média da amostra e o desvio padrão da amostra de 30 repetições.

Tabela B1 – Resultados para o problema de detecção de lesões mamária.

Modelos		Acur. (%)	Ind. Kappa	Sensibil.	Especific.	Precisão
DWNN2	SVM-Linear	66.4 (3.7)	0.33 (0.07)	0.47 (0.06)	0.86 (0.05)	0.77 (0.06)
	SVM-RBF1	54.7 (2.1)	0.09 (0.04)	0.10 (0.04)	0.99 (0.01)	0.96 (0.09)
	SVM-RBF2	66.3 (3.5)	0.33 (0.07)	0.45 (0.06)	0.88 (0.05)	0.79 (0.07)
	SVM-RBF3	66.0 (3.4)	0.32 (0.07)	0.43 (0.06)	0.88 (0.05)	0.79 (0.07)
	ELM-Linear	68.4 (6.7)	0.37 (0.13)	0.56 (0.12)	0.81 (0.05)	0.74 (0.08)
	ELM-P2	67.2 (5.8)	0.34 (0.12)	0.50 (0.08)	0.84 (0.08)	0.77 (0.09)
	ELM-P3	68.5 (5.4)	0.37 (0.11)	0.51 (0.07)	0.86 (0.08)	0.79 (0.08)
	ELM-P4	68.2 (5.8)	0.36 (0.12)	0.54 (0.08)	0.82 (0.09)	0.76 (0.08)
	ELM-P5	68.5 (5.5)	0.37 (0.11)	0.55 (0.08)	0.82 (0.09)	0.76 (0.08)
DWNN4	SVM-Linear	86.9 (2.9)	0.74 (0.06)	0.79 (0.05)	0.95 (0.03)	0.94 (0.03)
	SVM-RBF1	76.2 (3.3)	0.52 (0.07)	0.58 (0.06)	0.95 (0.03)	0.92 (0.05)
	SVM-RBF2	90.6 (2.7)	0.81 (0.05)	0.88 (0.04)	0.93 (0.03)	0.93 (0.03)
	SVM-RBF3	93.8 (2.2)	0.88 (0.04)	0.94 (0.03)	0.94 (0.03)	0.94 (0.03)
	ELM-Linear	85.9 (5.9)	0.72 (0.12)	0.79 (0.11)	0.93 (0.04)	0.92 (0.04)
	ELM-P2	89.1 (3.9)	0.78 (0.08)	0.82 (0.07)	0.96 (0.07)	0.96 (0.06)
	ELM-P3	85.9 (5.9)	0.72 (0.12)	0.77 (0.07)	0.95 (0.09)	0.95 (0.09)
	ELM-P4	86.9 (4.2)	0.74 (0.08)	0.78 (0.04)	0.96 (0.08)	0.95 (0.07)
	ELM-P5	85.2 (5.6)	0.70 (0.11)	0.76 (0.07)	0.95 (0.11)	0.95 (0.09)
DWNN6	SVM-Linear	99.0 (0.9)	0.98 (0.02)	0.98 (0.02)	1.00 (0.00)	1.00 (0.00)
	SVM-RBF1	99.3 (0.7)	0.99 (0.01)	0.99 (0.01)	1.00 (0.01)	1.00 (0.01)
	SVM-RBF2	93.9 (2.0)	0.88 (0.04)	1.00 (0.00)	0.88 (0.04)	0.89 (0.03)
	SVM-RBF3	93.9 (2.0)	0.88 (0.04)	1.00 (0.00)	0.88 (0.04)	0.89 (0.03)
	ELM-Linear	95.5 (4.8)	0.91 (0.10)	0.91 (0.09)	1.00 (0.01)	1.00 (0.01)
	ELM-P2	95.9 (4.0)	0.92 (0.08)	0.92 (0.08)	1.00 (0.01)	0.99 (0.01)
	ELM-P3	95.6 (3.6)	0.91 (0.07)	0.92 (0.07)	0.99 (0.02)	0.99 (0.02)
	ELM-P4	94.5 (5.3)	0.89 (0.11)	0.90 (0.08)	0.99 (0.03)	0.99 (0.03)
	ELM-P5	92.6 (5.3)	0.85 (0.11)	0.86 (0.09)	0.99 (0.02)	0.99 (0.02)

Fonte: O Autor (2022).

Tabela B2 – Resultados para o problema de classificação de lesões mamária.

Modelos		Acur. (%)	Ind. Kappa	Sensibil.	Especific.	Precisão
	SVM-Linear	50,1 (4,9)	0,25 (0,07)	0,81 (0,08)	0,54 (0,07)	0,47 (0,04)
	SVM-RBF1	37,1 (3,4)	0,06 (0,05)	0,77 (0,40)	0,34 (0,35)	0,37 (0,03)
	SVM-RBF2	49,9 (4,9)	0,25 (0,07)	0,85 (0,07)	0,51 (0,07)	0,46 (0,04)
	SVM-RBF3	49,7 (4,9)	0,25 (0,07)	0,82 (0,08)	0,53 (0,07)	0,47 (0,04)
DWNN2	ELM-Linear	49,6 (7,6)	0,24 (0,11)	0,50 (0,18)	0,75 (0,14)	0,52 (0,14)
	ELM-P2	50,8 (7,1)	0,26 (0,11)	0,51 (0,22)	0,75 (0,14)	0,52 (0,11)
	ELM-P3	51,0 (6,9)	0,26 (0,10)	0,51 (0,25)	0,75 (0,15)	0,52 (0,10)
	ELM-P4	51,1 (8,7)	0,27 (0,13)	0,51 (0,25)	0,76 (0,14)	0,52 (0,12)
	ELM-P5	51,3 (8,6)	0,27 (0,13)	0,51 (0,24)	0,76 (0,14)	0,53 (0,12)
	SVM-Linear	79,0 (4,8)	0,69 (0,07)	0,89 (0,06)	0,85 (0,05)	0,75 (0,07)
	SVM-RBF1	59,1 (4,4)	0,39 (0,07)	0,93 (0,05)	0,57 (0,07)	0,52 (0,04)
	SVM-RBF2	83,5 (4,3)	0,75 (0,06)	0,89 (0,06)	0,90 (0,04)	0,82 (0,06)
	SVM-RBF3	86,7 (3,8)	0,80 (0,06)	0,87 (0,07)	0,95 (0,03)	0,89 (0,06)
DWNN4	ELM-Linear	76,0 (15,4)	0,64 (0,23)	0,76 (0,19)	0,88 (0,09)	0,77 (0,16)
	ELM-P2	80,3 (11,8)	0,70 (0,18)	0,80 (0,16)	0,90 (0,09)	0,82 (0,13)
	ELM-P3	79,2 (12,9)	0,69 (0,19)	0,79 (0,16)	0,90 (0,09)	0,80 (0,14)
	ELM-P4	82,1 (9,7)	0,73 (0,15)	0,82 (0,13)	0,91 (0,07)	0,83 (0,12)
	ELM-P5	81,9 (10,5)	0,73 (0,16)	0,82 (0,15)	0,91 (0,06)	0,83 (0,12)
	SVM-Linear	97,3 (1,9)	0,96 (0,03)	1,00 (0,01)	0,97 (0,02)	0,95 (0,04)
	SVM-RBF1	95,6 (2,2)	0,93 (0,03)	0,97 (0,03)	0,97 (0,02)	0,95 (0,04)
	SVM-RBF2	79,6 (4,4)	0,69 (0,07)	0,70 (0,09)	1,00 (0,00)	1,00 (0,00)
	SVM-RBF3	78,7 (4,4)	0,68 (0,07)	0,70 (0,09)	1,00 (0,00)	1,00 (0,01)
DWNN6	ELM-Linear	92,5 (8,6)	0,89 (0,13)	0,92 (0,12)	0,96 (0,05)	0,93 (0,09)
	ELM-P2	91,8 (8,0)	0,88 (0,12)	0,92 (0,12)	0,96 (0,05)	0,92 (0,09)
	ELM-P3	94,2 (5,8)	0,91 (0,09)	0,94 (0,10)	0,97 (0,04)	0,95 (0,07)
	ELM-P4	92,2 (7,5)	0,88 (0,11)	0,92 (0,11)	0,96 (0,04)	0,93 (0,08)
	ELM-P5	89,9 (9,3)	0,85 (0,14)	0,90 (0,14)	0,95 (0,06)	0,90 (0,10)

Fonte: O Autor (2022).