

**UNIVERSIDADE NOVE DE JULHO - UNINOVE  
PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA E GESTÃO  
DO CONHECIMENTO - PPGI**

**Guilherme Bueno Martins**

**Uma ferramenta baseada em Redes Morfológicas Profundas para auxiliar no  
diagnóstico de hipertrofia adenoideana**

**São Paulo  
2023**

**Guilherme Bueno Martins**

**Uma ferramenta baseada em Redes Morfológicas Profundas para auxiliar no diagnóstico de hipertrofia adenoideana**

Exame de Defesa apresentado a Universidade Nove de Julho - UNINOVE, como parte dos requisitos para a obtenção do título de Mestre em Informática e Gestão do Conhecimento.

Prof. Orientador: Dr. Wonder Alexandre Luz Alves

**São Paulo  
2023**

Martins, Guilherme Bueno.

Uma ferramenta baseada em redes morfológicas profundas para auxiliar no diagnóstico de hipertrofia adenoideana. /

Guilherme Bueno Martins. 2023.

56 f.

Dissertação (Mestrado) - Universidade Nove de Julho - UNINOVE, São Paulo, 2023.

Orientador (a): Prof. Dr. Wonder Alexandre Luz Alves.

1. Redes morfológicas profundas. 2. Redes neurais artificiais. 3. Visão computacional. 4. Hipertrofia adenoideana. 5. Otorrinolaringologia.

I. Alves, Wonder Alexandre Luz.

CDU 004

## PARECER – EXAME DE DEFESA

Parecer da Comissão Examinadora designada para o exame de defesa do Programa de Pós-Graduação em Informática e Gestão do Conhecimento a qual se submeteu o aluno Guilherme Bueno Martins.

Tendo examinado o trabalho apresentado para obtenção do título de “Mestre em Informática e Gestão do Conhecimento”, com Dissertação intitulada “Uma ferramenta baseada em Redes Morfológicas Profundas para auxiliar no diagnóstico de hipertrofia adenoideana”, a Comissão Examinadora considerou o trabalho:

**(X) Aprovado**

**( ) Aprovado condicionalmente**

**( ) Reprovado com direito a novo exame**

**( ) Reprovado**

## EXAMINADORES

Prof. Dr. Wonder Alexandre Luz Alves / Uninove (Orientador)

Dr. Wonder Alexandre Luz Alves - Uninove (

Prof. Dr. Daniel Oliveira Dantas - UFS (Membro Externo)

Handwritten signatures: [Illegible]

Prof. Dr. Sidnei Alves de Araújo / UNINOVE (Membro Interno)

Alves de Araújo - U

São Paulo, 29 de junho de 2023

A aprendizagem de máquina revolucionou diversos campos do conhecimento, incluindo a área médica, ao melhorar significativamente o estado da arte de diversas aplicações. Uma aplicação de destaque é o processamento de imagens médicas utilizando redes neurais convolucionais. No entanto, a escassez de amostras ou dados médicos disponíveis dificulta a aplicação efetiva dessa técnica. Neste trabalho é proposto um método baseado em redes morfológicas profundas para a elaboração de uma ferramenta para software livre a fim de auxiliar no diagnóstico da hipertrofia adenoideana. A ferramenta utiliza redes morfológicas profundas para reconhecer quais frames (imagens) de um vídeos de exame de nasofibrolaringoscopia são ideias para o médico especialista determinar um diagnóstico de hipertrofia adenoideana. O processo de construção dessa ferramenta envolve as seguintes etapas: (1) Criação de um banco de exames de nasofibrolaringoscopia; (2) Criação de um *dataset* rotulado por um especialista; (3) Criação de um modelo baseado em redes morfológicas profundas; (4) Criação de um *plugin* para software livre ImageJ, utilizando o modelo proposto, que foi devidamente treinado e validado. O melhor modelo avaliado combina a arquitetura LeNet com operadores morfológicos, sua acurácia é superior a 95% com uma precisão acima de 90%. Por meio desse trabalho, espera-se contribuir para o avanço no diagnóstico de hipertrofia adenoideana, fornecendo aos médicos uma ferramenta eficaz e confiável baseada em aprendizagem de máquina.

**Palavras-chave:** Redes Morfológicas Profundas; Redes Neurais Artificiais; Visão Computacional; Hipertrofia adenoideana; Otorrinolaringologia

## ABSTRACT

---

Machine learning has revolutionized various fields of knowledge, including the medical field, significantly improving the state of the art in numerous applications. One prominent application is the processing of medical images using convolutional neural networks. However, the scarcity of available medical samples or data poses challenges to the effective application of this technique. This work proposes a method based on deep morphological networks for the development of an open-source software-based tool that aids in the diagnosis of adenoid hypertrophy. The tool employs machine learning techniques to recognize whose frames (images) in a video of nasopharyngolaryngoscopy exam are ideals to the specialist medical formulate a adenoid hypertrophy diagnosis. The construction process of this tool involves the following steps: (1) Creation of a nasopharyngolaryngoscopy exam database. (2) Creation of a dataset labeled by a specialist. (3) Development of a model based on deep morphological networks. (4) Creation of a plugin for the open-source software ImageJ, utilizing the proposed model, which has been appropriately trained and validated. The proposed model combines the LeNet architecture with morphological operators, its accuracy is superior to 90% and precision above 90%. Through this work, we aim to contribute to the advancement of adenoid hypertrophy diagnosis, providing medical professionals with an effective and reliable machine learning-based tool.

**Keywords:** Deep Morphological Networks; Artificial Neural Networks, Computational Vision, Adenoid Hypertrophy, Otolaryngology

---

|   |           |
|---|-----------|
| <b>Lista de Figuras</b>   | <b>7</b>  |
| <b>Lista de Símbolos</b>  | <b>8</b>  |
| <b>1 Introdução</b>   | <b>10</b> |
| 1.1 Contextualização . . . . .  | 10        |
| 1.2 Justificativa e problema de pesquisa . . . . .                            | 11        |
| 1.3 Objetivos . . . . .   | 12        |
| <b>2 Conceitos sobre imagens</b>  | <b>14</b> |
| 2.1 Imagem . . . . .  | 14        |
| 2.2 Operadores de imagem . . . . .  | 15        |
| 2.3 Elementos estruturantes . . . . .   | 16        |
| 2.4 Operadores morfológicos elementares de imagens em níveis de cinza . . . . | 17        |
| 2.5 Gradientes morfológicos e top-hats . . . . .                              | 18        |
| 2.6 Operador de convolução . . . . .  | 19        |
| <b>3 Conceitos sobre aprendizagem profunda</b>                                | <b>22</b> |
| 3.1 Introdução à aprendizagem de máquina supervisionada . . . . .             | 22        |
| 3.2 Redes neurais artificiais . . . . .                                       | 23        |
| 3.2.1 Funções de ativação . . . . .   | 25        |
| 3.2.2 Propagação para frente . . . . .  | 26        |
| 3.2.3 Função de custo . . . . .   | 28        |
| 3.2.4 Retropropagação . . . . .   | 29        |
| 3.3 Redes neurais convolucionais . . . . .                                    | 30        |
| 3.4 Redes neurais morfológicas . . . . .                                      | 33        |
| 3.5 Regularização e otimização . . . . .                                      | 35        |
| 3.6 Hiperparâmetros . . . . .   | 36        |
| <b>4 Trabalhos relacionados</b>   | <b>37</b> |
| 4.1 Trabalhos endoscópicos . . . . .  | 37        |
| 4.2 Outros trabalhos . . . . .  | 39        |
| <b>5 Método proposto</b>  | <b>42</b> |
| 5.1 Criação de uma banco de exames de nasofibrolaringoscopia . . . . .        | 43        |
| 5.2 Criação de um dataset rotulado . . . . .                                  | 43        |
| 5.3 Criação de um modelo baseado em redes morfológicas profundas . . . . .    | 45        |
| 5.4 Plugin ImageJ . . . . .   | 46        |

|  |           |
|--|-----------|
| SUMÁRIO  | 6         |
| <b>6 Resultados</b>  | <b>48</b> |
| 6.1 Divisão das amostras para treinamento, validação e teste . . . . . | 48        |
| 6.2 Seleção de modelos e otimização dos hiperparâmetros . . . . .      | 49        |
| 6.3 Resultados dos modelos selecionados . . . . .                      | 51        |
| 6.4 Análise dos resultados . . . . .                                   | 52        |
| <b>7 Conclusão</b>   | <b>53</b> |
| <b>Referências Bibliográficas</b>                                      | <b>54</b> |



## LISTA DE FIGURAS

---

|      |   |    |
|------|---|----|
| 2.1  | Exemplos de relações de adjacências . . . . .   | 15 |
| 2.2  | Exemplo de extração de componentes conexos . . . . .  | 15 |
| 2.3  | Exemplo de extração dos CCs do conjunto $\mathcal{X} \subseteq \mathcal{D}$ sobre a adjacência $\mathcal{A}_8$ . . . .  | 15 |
| 2.4  | Exemplos de EEs planos e não planos com suas respectivas origens indicadas em vermelho: (a) EE plano coluna; (b) EE plano linha; (c) EE plano cruz; (d) EE não plano coluna; (e) EE não plano linha; e (f) EE não plano cruz. . . . . | 16 |
| 2.5  | Exemplo da aplicação de erosão e dilatação com EE formato de cruz . . . . .   | 17 |
| 2.6  | Exemplo da aplicação de erosão e dilatação com EE volumético formato de cruz  | 18 |
| 2.7  | Exemplo de aplicação de abertura e fechamento com EE formato de cruz em uma dada imagem $f$ . . . . .   | 18 |
| 2.8  | Exemplo de aplicação de gradiente e gradiente interno com EE formato de cruz em uma dada imagem $f$ . . . . .   | 19 |
| 2.9  | Exemplo de aplicação de white top-hat e black top-hat com EE formato de cruz em uma dada imagem $f$ . . . . .   | 19 |
| 2.10 | Convolução etapa a etapa de uma imagem $f$ . . . . .  | 21 |
| 3.1  | Processo de geração da função hipótese a partir de um algoritmo de treinamento.   | 23 |
| 3.2  | Representação visual de um neurônio artificial do tipo Perceptron. . . . .  | 24 |
| 3.3  | Representação visual de uma MLP com uma camada escondida. . . . .   | 24 |
| 3.4  | Modelo abstrado de uma rede neural artificial. . . . .  | 27 |
| 3.5  | Modelo abstrato de um rede neural convolucional . . . . .   | 31 |
| 3.6  | Exemplo de operação <i>max pooling</i> aplicado em uma imagem $f$ . . . . .   | 32 |
| 3.7  | Modelo abstrato de uma rede neural morfológica. . . . .   | 33 |
| 5.1  | Visão geral do método proposto . . . . .  | 42 |
| 5.2  | Diagrama de processos da metodologia para rotulação dos <i>frames</i> de interesses dos vídeos dos exames de nasofibrolaringoscopia. . . . .  | 44 |
| 5.3  | Exemplo de <i>frames</i> rotulados como negativo e positivo por especialistas. . . . .  | 44 |
| 5.4  | Arquitetura proposta da rede neural artificial . . . . .  | 46 |
| 5.5  | Apresentação do menu <i>Plugin</i> do software ImageJ. . . . .  | 46 |
| 5.6  | Interface implementação <i>PlugIn</i> do ImageJ. . . . .  | 47 |
| 5.7  | Ilustração do resultado encontrado pelo plugin no ImageJ . . . . .  | 47 |
| 6.1  | Curvas ROC dos modelos . . . . .  | 52 |

### CONCEITOS BÁSICOS

|                          |   |
|--------------------------|---|
| $\mathbb{N}$             | Conjunto dos números naturais                     |
| $\mathbb{Z}$             | Conjunto dos números inteiros                     |
| $\mathbb{R}$             | Conjunto dos números reais                        |
| $\mathbb{R}_{\pm\infty}$ | Conjunto dos elementos afinadamente estendidos    |
| $-\infty$                | Representação do menor valor possível em conjunto |
| $+\infty$                | Representação do maior valor possível em conjunto |
| $\in$                    | Pertence ao conjunto                              |
| $\notin$                 | Não pertence ao conjunto                          |
| $\subset$                | É subconjunto de                                  |
| $\forall$                | Quantificador lógico universal                    |
| $\exists$                | Quantificador lógico existencial                  |
| $\max\{\}$               | Elemento máximo de um conjunto                    |
| $\min\{\}$               | Elemento mínimo de um conjunto                    |

## CONCEITOS PRELIMINARES

|                             |  |
|-----------------------------|--|
| $f$                         | Função que representa imagem   |
| $\mathcal{D}$               | Conjunto de <i>pixels</i> que representa o domínio da imagem               |
| $\mathcal{I}$               | Conjunto de <i>pixels</i> que representa o contradomínio da imagem         |
| $I$                         | Valor de nível de cinza máximo do conjunto $\mathcal{I}$                   |
| $k$                         | Valor dos bits de profundidade   |
| $\mathcal{X}$               | Conjunto de <i>pixels</i> de $f$ de nível de cinza 1                       |
| $p, q \in \mathcal{D}$      | Pares ordenados (x, y) que represem <i>pixels</i>                          |
| $f(p)$                      | Valor de nível de cinza do <i>pixel</i> $p$ pertencente a $f$              |
| $\mathcal{A}_4$             | Relação de adjacência <i>vizinhança-4</i> sobre $\mathcal{D}$              |
| $\mathcal{A}_8$             | Relação de adjacência <i>vizinhança-8</i> sobre $\mathcal{D}$              |
| $\mathcal{F}(\mathcal{D})$  | Conjunto de imagens de domínio $\mathcal{D}$ e contradomínio $\mathcal{I}$ |
| $\psi, \theta$              | Representação de operadores de imagens genéricos                           |
| $f^c$                       | Complemento da imagem $f$  |
| $\mathcal{B}$               | Conjunto que representa elemento estruturante                              |
| $B(b)$                      | Valor do elemento do conjunto $B$  |
| $\mathcal{A}_{\mathcal{B}}$ | Adjacência de um elemento estruturante $B$                                 |
| $\delta_{\mathcal{B}}(f)$   | Operação de dilatação de $f$ a partir de $\mathcal{B}$                     |
| $\epsilon_{\mathcal{B}}(f)$ | Operação de erosão de $f$ a partir de $\mathcal{B}$                        |
| $\gamma_{\mathcal{B}}(f)$   | Operação de abertura de $f$ a partir de $\mathcal{B}$                      |
| $\phi_{\mathcal{B}}(f)$     | Operação de fechamento de $f$ a partir de $\mathcal{B}$                    |
| $\nabla_{\mathcal{B}}(f)$   | Operação de gradiente de $f$ a partir de $\mathcal{B}$                     |
| $\nabla_{\mathcal{B}}^-(f)$ | Operação de gradiente interno de $f$ a partir de $\mathcal{B}$             |
| $\rho_{\mathcal{B}}^+(f)$   | Operação de white top-hat de $f$ a partir de $\mathcal{B}$                 |
| $\rho_{\mathcal{B}}^-(f)$   | Operação de black top-hat de $f$ a partir de $\mathcal{B}$                 |

## 1.1 CONTEXTUALIZAÇÃO

Os modelos computacionais de aprendizagem profunda melhoraram drasticamente o estado da arte em reconhecimento de objetos em imagens bem como em muitos outros domínios do conhecimento. Estes modelos compostos de múltiplas camadas de processamento permitindo aprendizagem de representações de dados (LECUN; BENGIO; HINTON, 2015) são de interesse crescente para médicos pesquisadores e profissionais dos setores da saúde. A capacidade de processar muito mais informações que seres humanos e a possibilidade do aumento da acurácia em aplicações médicas (PICCIALLI et al., 2021) tem sido um dos fatores pelo qual a inteligência artificial, como os algoritmos de aprendizagem profunda, vem ganhando notoriedade (ESTEVA et al., 2021). Uma das técnicas de inteligência artificial bastante utilizada para processamento de imagens médicas são as redes neurais convolucionais (PICCIALLI et al., 2021). Técnica que consiste na aplicação de operações lineares chamadas convoluções e que tem apresentado sucesso em suas aplicações práticas (GOODFELLOW; BENGIO; COURVILLE, 2016). Mas a quantidade limitada de amostras ou dados médicos dificultam a aplicação desta técnica de aprendizagem profunda (GIRDLER et al., 2021; ZHENG et al., 2022; SHEN et al., 2020).

A forma de objetos é uma característica visual importante e básica para descrever conteúdo de imagens. Na representação de um objeto 3D do mundo real em 2D de superfície plana, uma dimensão das informações do objeto é perdida dificultando a tarefa de descrição e representação de formas. Pois, a forma extraída de imagens apenas parcialmente representa a projeção do objeto (ZHANG; LU, 2004). Processos de transformações lineares, como a convolução, podem borrar arestas e prejudicar noções de bordas e curvas em ponderações dos píxeis da imagem (NOGUEIRA et al., 2019) desejando-se operadores não lineares como os filtros morfológicos (SERRA; SOILLE, 1994). Estes operadores não lineares voltados para o processamento de imagem providos pela Morfologia Matemática (FRANCHI; FEHRI; YAO, 2020) são considerados estado da arte para algumas aplicações (NOGUEIRA et al., 2019) e de crescente interesse (MONDAL; DEY; CHANDA, 2020; KIRSZENBERG et al., 2021).

A Morfologia Matemática é uma área de estudos focada em analisar a estrutura e forma dos objetos fundamentada em operações matemáticas como operações algébricas e de conjuntos (SERRA; SOILLE, 1994). Voltada para tarefas de processamento de imagens como filtragem, segmentação e detecção de arestas, dentre outras (FRANCHI; FEHRI; YAO, 2020). A aprendizagem de filtros morfológicos em redes neurais morfológicas profundas (KIRSZENBERG et al., 2021; MONDAL; DEY; CHANDA, 2020; MASCI; ANGULO; SCHMIDHUBER, 2013) é uma técnica promissora e útil no processamento de imagens de

sensoriamento remoto e outras aplicações. Isto inclui o uso de redes neurais morfológicas para o auxílio no diagnóstico de hipertrofia adenoideana a partir da seleção de imagens de exame.

## 1.2 JUSTIFICATIVA E PROBLEMA DE PESQUISA

A hipertrofia adenoideana, nome dado ao crescimento excessivo da tonsila faríngea, pode causar sintomas como ronco, apneia, fadiga e o impedimento da respiração durante o sono, entre outros sintomas. Sendo diagnóstico comum em crianças entre 4 a 7 anos de idade com obstrução de vias áreas superiores (MAAHS; ALMEIDA, 2017; PIGNATARI; ANSELMO-LIMA, 2017; YSUNZA et al., 2008; SHEN et al., 2020; BALDASSARI; CHOI, 2014). Ela é frequentemente diagnosticada pela nasofibrolaringoscopia (ou videonasofibrolaringoscopia) (MAAHS; ALMEIDA, 2017; PIGNATARI; ANSELMO-LIMA, 2017; SHEN et al., 2020), um exame considerado padrão ouro (PILTCHER et al., 2014) e realizado por otorrinolaringologistas. Esse exame consiste na introdução do aparelho de nasofibrolaringoscopia pela fossa nasal mais ampla do paciente enquanto um vídeo é gravado pelo instrumento. Posteriormente, esse vídeo é analisado por um otorrinolaringologista para elaborar um diagnóstico do exame.

Conforme aponta Baldassari e Choi (2014) e Pignatari e Anselmo-Lima (2017), o custo financeiro para realizar o exame da nasofibrolaringoscopia é tão barato quanto o exame da radiografia nasal lateral. No entanto, a nasofibrolaringoscopia sendo um exame dinâmico e sensível (PIGNATARI; ANSELMO-LIMA, 2017; BALDASSARI; CHOI, 2014; MAAHS; ALMEIDA, 2017), pode contemplar etapas de exames que visam avaliar o aspecto da mucosa, presença de desvios septais, anatomia das conchas nasais, meatos e recessos, tornando-a o padrão ouro na avaliação de vias aéreas superiores (PIGNATARI; ANSELMO-LIMA, 2017; YSUNZA et al., 2008). No quesito segurança, é importante ressaltar que a não exposição a radiação é um fator importante (BALDASSARI; CHOI, 2014) que possibilita o especialista repetir o exame no paciente. Quando pacientes crianças são submetidos ao exame, percebe-se na interpretação de Baldassari e Choi (2014) a necessidade de anestesia tópica. Isto para se obter um vídeo com uma melhor qualidade, dado que o exame é gravado (PIGNATARI; ANSELMO-LIMA, 2017).

A inevitável agitação ou falta de colaboração das crianças durante a realização do exame (YSUNZA et al., 2008; BALDASSARI; CHOI, 2014) propicia gravações mais longas, imagens agitadas ou com ruídos que prejudicam a análise da gravação pelo especialista. O diagnóstico do médico otorrinolaringologista é dependente da qualidade das imagens e da busca do *frame* (quadro ou imagem) do vídeo mais adequada para estabelecer o diagnóstico. Devido ao impacto que a qualidade das imagens bem como o tempo total do vídeo exerce na seleção dos *frames* para diagnóstico, foi levantada a seguinte pergunta:

- **A partir da gravação feita por otorrinolaringologista, como selecionar de forma automática as imagens de interesse?**

Sendo a gravação do especialista um conjunto de *frames* ou imagens que compõe o vídeo de nasofibrolaringoscopia, selecionar de forma automática as imagens de interesse é usar uma ferramenta ou procedimento mecanizado para a classificação destes *frames* partindo do critério de que a imagem escolhida deve permitir ao especialista fazer o diagnóstico de hipertrofia adenoideana. Esta ferramenta capaz de classificar os *frames* em imagens de interesse ou não foi construída a partir de um método que envolve as técnicas de redes neurais convolucionais e redes neurais morfológicas. As redes neurais convolucionais como uma técnica que vem apresentando sucesso no processamento de imagens (GOOD-FELLOW; BENGIO; COURVILLE, 2016), incluindo imagens médicas (PICCIALLI et al., 2021), será útil como ponto de partida. E as redes neurais morfológicas serão bem úteis para minimizar os impactos da quantidade limitada de informações por meio de uma melhor extração de características das imagens e, por conseguinte, melhores resultados obtidos com as redes neurais convolucionais. Este método descreve os processos de desenvolvimento da ferramenta para a classificação dos *frames*. A ferramenta estará disponível ao médico especialista por meio de um plugin do software livre *ImageJ*, idealizado pelo National Institute of Health, para auxiliá-lo no diagnóstico selecionando os melhores frames do vídeo do exame para o diagnóstico de hipertrofia adenoideana. Como um software livre de domínio público, desenvolvido em Java, capaz de editar, processar e analisar imagens e vídeos e, disponível para os sistemas operacionais Linux, MacOS e Windows, proporciona uma boa acessibilidade. Com uma arquitetura aberta e expansível por meio de *plugins* e macros, o *ImageJ* adequa-se a finalidade de disponibilizar a ferramenta a pesquisadores e profissionais da área médica.

### 1.3 OBJETIVOS

Conforme a pergunta de pesquisa, tem-se como objetivo desta pesquisa projetar uma ferramenta baseada em redes neurais morfológicas para auxiliar no diagnóstico de hipertrofia adenoideana. Assim constituem-se como objetivos específicos:

- 1) Construir um modelo baseado em redes morfológicas profundas para selecionar imagens ideais para classificação de hipertrofia adenoideana extraídas de vídeos de exame de nasofibrolaringoscopia;
- 2) Validar o método proposto utilizando um banco de exames de nasofibrolaringoscopia anotado por especialistas;
- 3) Desenvolver um *plugin* para software livre *ImageJ* para extração de imagens de interesse a partir do vídeo do exame de nasofibrolaringoscopia. Esse *plugin*, que é destinado

a otorrinolaringologistas, faz uso dos modelos de redes morfológicas profundas desenvolvidos nessa pesquisa.

## CONCEITOS SOBRE IMAGENS

**Resumo do capítulo**

*Este capítulo descreve conceitos básicos possíveis de serem encontrados em livros de que tratam de análise e processamento de imagens com o propósito de facilitar o entendimento deste trabalho. Os conceitos e definições apresentados neste capítulo foram obtidos em livros como (SOILLE et al., 1999), (DOUGHERTY; LOTUFO, 2003) e (GOODFELLOW; BENGIO; COURVILLE, 2016)*

**2.1 IMAGEM**

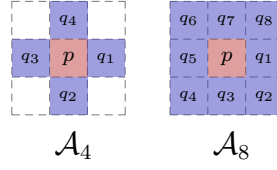
Uma imagem  $f$  é um mapeamento finito e retangular de domínio  $\mathcal{D} \subset \mathbb{Z} \times \mathbb{Z}$  para o conjunto discreto  $\mathcal{I} = \{0, \dots, \mathcal{I}_{\max}\}$ . O conjunto de níveis de cinza  $\mathcal{I} \subset \mathbb{Z}$  contém valor máximo de nível de cinza  $\mathcal{I}_{\max} = 2^k - 1$  onde  $k > 0$  representa a quantidade de bits de profundidade da imagem. Um elemento do domínio da imagem é chamado de pixel, ou seja,  $p = (p_x, p_y) \in \mathcal{D}$  é um pixel cujo os valores  $p_x$  e  $p_y$  são os valores das coordenadas horizontal e vertical de  $p$ . O nível de cinza de um pixel  $p$  na imagem  $f$  é dado por  $f(p)$ .

Diz-se que uma imagem  $f$  é binária quando  $k = 1$ . Nesse caso, a imagem tem dois tipos de pixels: os que pertencem ao objeto da imagem, *foreground*, cujo nível de cinza é 1 e denota-se por  $\mathcal{X} = \{p \in \mathcal{D} : f(p) = 1\}$  e os que pertencem ao fundo da imagem, *background*, cujo nível de cinza é 0. Analogamente, uma imagem em tons de cinza (ou níveis de cinza) é quando  $k > 1$ . Ademais, imagens multiespectrais (ou multibandas) são vetores de imagens  $(f_1, f_2, \dots, f_n)$  com quantidade  $n$  de bandas espectrais. As imagens coloridas em RGB são exemplos de imagens multibandas formadas pelas bandas vermelha, verde e azul.

Um pixel está conectado a outros pixels em seu entorno e este conceito de conectividade (ou vizinhança) entre os pixels é definido por meio de uma relação de adjacência. Mais precisamente, uma relação de adjacência  $\mathcal{A}$  é uma relação binária estabelecida entre pixels, ou seja  $\mathcal{A} \subseteq \mathcal{D} \times \mathcal{D}$ . Assim, é dito que  $p$  é adjacente (vizinho) a  $q$  se  $(p, q) \in \mathcal{A}$  com  $p \neq q$ . A título de exemplo, considerando uma relação de adjacência circular e simétrica definida pela distância euclidiana  $d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$ , têm-se que  $\mathcal{A} = \{(p, q) : p, q \in \mathcal{D}, d(p, q) \leq r\}$ , onde o raio  $r$  delimita a adjacência (vizinhança) dos pixels. Dessa forma, se  $r = 1$ , temos a relação de adjacência, denotada por  $\mathcal{A}_4$ , que define a conhecida vizinhança-4 formados pelos pixels horizontais e verticais em relação ao pixel de referência. Analogamente se  $r = \sqrt{2}$ , temos o caso da vizinhança-8, denotada por  $\mathcal{A}_8$ , formados pelos pixels horizontais, verticais e diagonais.

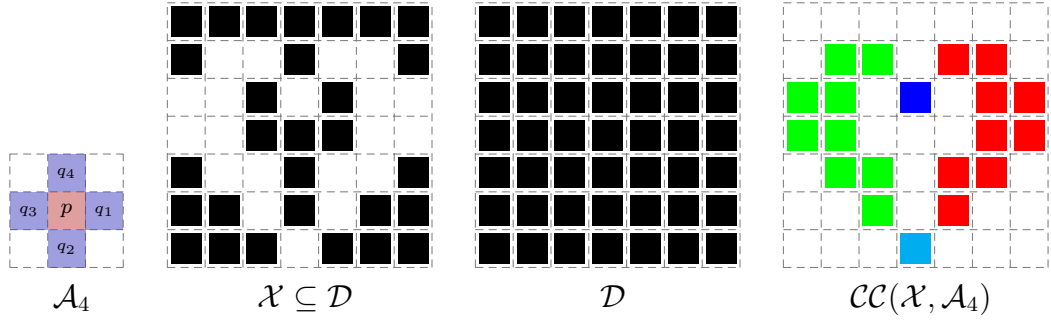
Chama-se de componente conexo (CC)  $\mathcal{C}$  um subconjunto maximal de pixels de uma



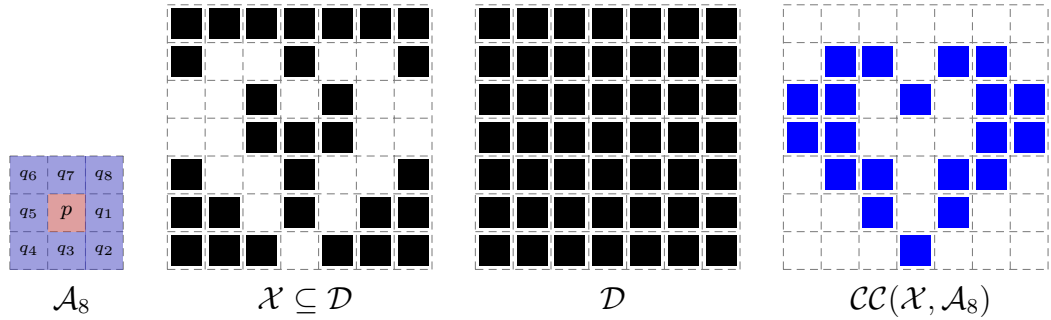


**Figura 2.1:** Exemplos de relações de adjacências para um dado pixel  $p$ .

imagem binária  $\mathcal{X}$  em que para quaisquer dois pixels  $p$  e  $q \in \mathcal{X}$  existe um caminho  $(p_1, p_2, \dots, p_n)$  de  $p$  até  $q$  em  $\mathcal{C}$  tal que  $p = p_1$ ,  $q = p_n$  e  $p_i \in \mathcal{A}(p_i)$  com  $0 < i \leq n$ .



**Figura 2.2:** Exemplo de extração dos CCs do conjunto  $\mathcal{X} \subseteq \mathcal{D}$  sobre a adjacência  $\mathcal{A}_4$ .



**Figura 2.3:** Exemplo de extração dos CCs do conjunto  $\mathcal{X} \subseteq \mathcal{D}$  sobre a adjacência  $\mathcal{A}_8$ .

## 2.2 OPERADORES DE IMAGEM

Um operador de imagem em níveis de cinza  $\psi$  é um mapeamento de  $\mathcal{D}_{\mathcal{F}}$ , conjunto de todas as imagens em níveis de cinza de domínio  $\mathcal{D}$ , em  $\mathcal{D}_{\mathcal{F}}$ . Eles podem ser caracterizados através de suas propriedades, as quais são fundamentais para prever os aspectos do seu comportamento:

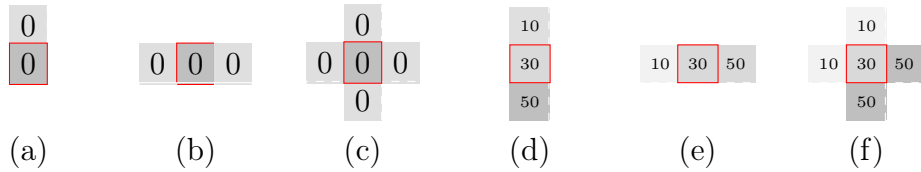
- *Idempotência:* É dito que um operador  $\psi$  possui idempotência quando o resultado de sua aplicação recorrente se equivale a uma única aplicação, ou seja,  $\psi(\psi(f)) = \psi(f)$ .
- *Extensividade:* Um operador é extensivo quando  $\psi$  aplicado a uma imagem  $f$  resulta em  $\psi(f) \geq f$ .

- *Anti-Extensividade*: Um operador é anti-extensivo quando  $\psi$  aplicado a uma imagem  $f$  resulta em  $\psi(f) \leq f$ .
- *Crescente*: Um operador é crescente quando ele preserva a relação de ordem que pode estar presente em duas imagens, isto é, para quaisquer imagens  $f$  e  $g$ , com  $f \geq g$ , a aplicação do operador  $\psi$  preserva a relação como  $\psi(f) \geq \psi(g)$ .
- *Dualidade*: Dois operadores  $\psi$  e  $\phi$  são duais com respeito a complementaridade se, ao aplicar  $\psi$  em uma imagem  $f$  é equivalente a aplicar  $\phi$  no negativo de  $f$ . O negativo de  $f$  é dado por  $f^c = \mathcal{I}_{\max} - f$ .

### 2.3 ELEMENTOS ESTRUTURANTES

Diversos operadores da Morfologia Matemática usam como base o elemento estruturante (EE) e estes podem ser planos ou não planos. Os EEs compreendidos como não planos, volumétricos, ou ainda, em níveis de cinza são definidos por meio de uma função estruturante  $\mathcal{B} : \mathcal{D}_{\mathcal{B}} \subseteq \mathbb{Z}^2 \rightarrow \mathbb{R}$  e eles são o foco dessa pesquisa. Já os EEs conhecidos como planos tem seus valores definidos como zero para todas as coordenadas do domínio de  $\mathcal{B}$  e por isso eles são representados simplesmente por um conjunto. Na Figura 2.4 são apresentados alguns exemplos de EEs.

Os EEs compreendidos como *não planos*, *volumétricos*, ou ainda, *em níveis de cinza* são aqueles que, para qualquer coordenada pertencente ao conjunto  $\mathcal{B}$ , existe um valor definido no conjunto dos números afinamente estendidos  $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{-\infty, +\infty\}$ .



**Figura 2.4:** Exemplos de EEs planos e não planos com suas respectivas origens indicadas em vermelho: (a) EE plano coluna; (b) EE plano linha; (c) EE plano cruz; (d) EE não plano coluna; (e) EE não plano linha; e (f) EE não plano cruz.

Assim aplicação das operações morfológicas ocorre a partir da translação da imagem  $f$  sobre o domínio do EE  $\mathcal{B}$ , isto é, novos valores de níveis de cinza são obtidos para cada pixel  $p \in \mathcal{D}$  da imagem  $f$  pela translação de  $p$  pelo domínio do EE  $\mathcal{B}$  dos operadores morfológicos definidos nas próximas seções.

## 2.4 OPERADORES MORFOLÓGICOS ELEMENTARES DE IMAGENS EM NÍVEIS DE CINZA

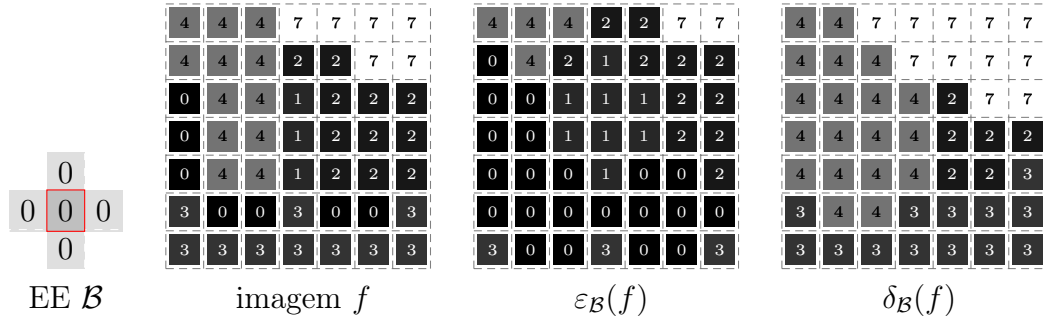
Os operadores morfológicos *erosão*, *dilatação*, *abertura* e *fechamento* são operadores básicos, mas os operadores erosão e dilatação são os primeiros e principais dentre eles. A erosão de uma imagem  $f$  a partir de um EE  $\mathcal{B}$  é representada  $\epsilon_{\mathcal{B}}(f)$ . O operador erosão em imagens de níveis de cinza caracteriza-se pela obtenção do valor mínimo definido pela translação da imagem sobre o EE, ou seja

$$\forall p \in \mathcal{D}, [\epsilon_{\mathcal{B}}(f)](p) = \min_{q \in \mathcal{B}(\mathcal{D})} \{f(p+q) - \mathcal{B}(q)\}. \quad (2.1)$$

Por sua vez, a operação de dilatação em uma imagem  $f$  em níveis de cinza a partir de um EE  $\mathcal{B}$  é obtida por meio dos valores máximo restritos pelo domínio do EE. Denotada como  $\delta_{\mathcal{B}}(f)$ , define-se a dilatação como

$$\forall p \in \mathcal{D}, [\delta_{\mathcal{B}}(f)](p) = \max_{q \in \mathcal{B}(\mathcal{D})} \{f(p-q) + \mathcal{B}(q)\}. \quad (2.2)$$

Conforme previamente mencionadas as propriedades dos operadores de imagens, as operações de erosão e dilatação são crescentes e duais entre si. Quando considerado EE de simetria radial (em relação a origem), temos  $\epsilon_{\mathcal{B}}(f)$  como um operador anti-extensivo, enquanto  $\delta_{\mathcal{B}}(f)$  como extensivo.

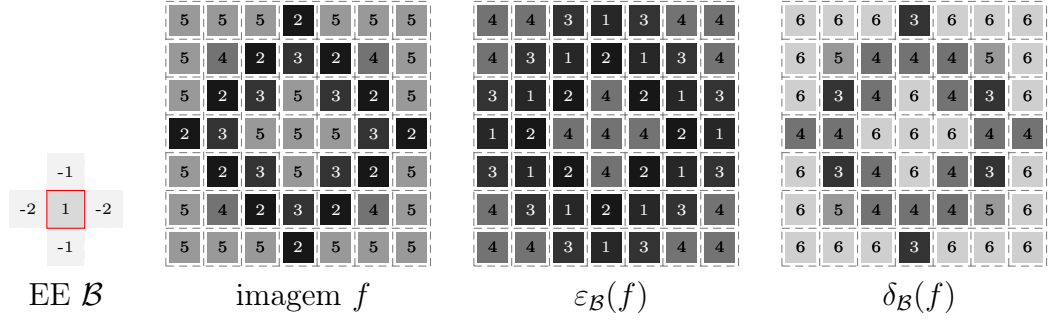


**Figura 2.5:** Exemplo da aplicação de erosão e dilatação com EE plano formato de cruz em uma dada imagem  $f$ .

Os operadores morfológicos abertura e fechamento são oriundos da combinação dos operadores de erosão e dilatação em uma dada sequência. Quando aplicada a dilatação a uma imagem resultante de uma erosão de mesmo EE  $\mathcal{B}$ , temos a aplicação da operação abertura denotada por  $\gamma_{\mathcal{B}}(f)$  e, analogamente, aplicada a dilatação seguida da erosão, tem-se a operação de fechamento representada por  $\phi_{\mathcal{B}}(f)$ , ou seja, Equações 2.3 e 2.4.

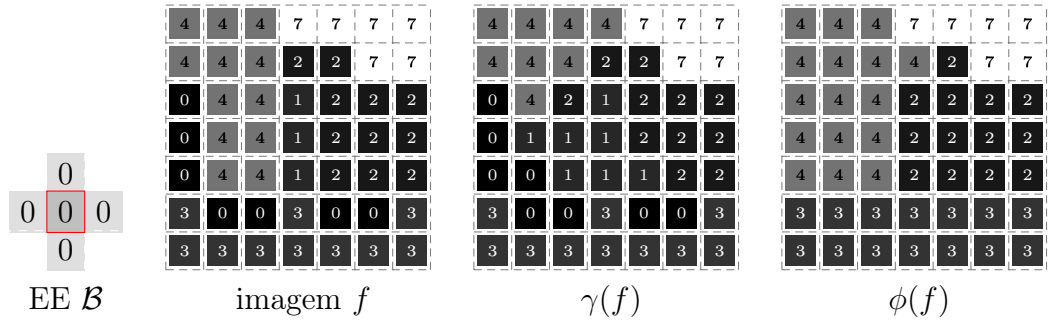
$$\gamma_{\mathcal{B}}(f) = \delta_{\mathcal{B}}(\epsilon_{\mathcal{B}}(f)). \quad (2.3)$$

$$\phi_{\mathcal{B}}(f) = \epsilon_{\mathcal{B}}(\delta_{\mathcal{B}}(f)). \quad (2.4)$$



**Figura 2.6:** Exemplo da aplicação de erosão e dilatação com EE volumétrico formato de cruz em uma dada imagem  $f$ .

Ambos, abertura e fechamento, possuem as mesmas propriedades que os operadores morfológicos que os compõem. Eles são crescentes, possuem uma relação de dualidade entre um e outro e, quando seus elementos estruturantes são radialmente simétricos, são operadores respectivamente extensivos e anti-extensivos. Além disso, são idempotentes.



**Figura 2.7:** Exemplo de aplicação de abertura e fechamento com EE formato de cruz em uma dada imagem  $f$ .

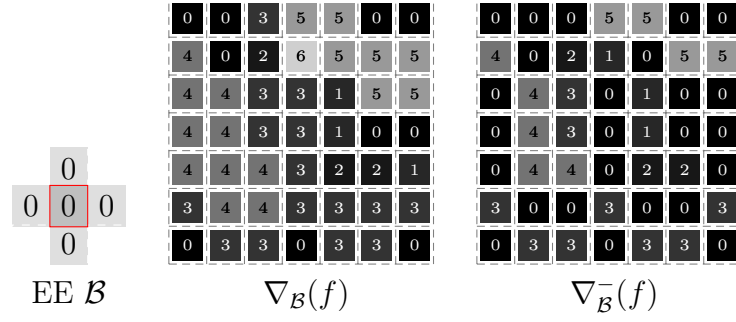
## 2.5 GRADIENTES MORFOLÓGICOS E TOP-HATS

Os *gradientes morfológicos* e operadores *top-hats* são definidos a partir dos conceitos dos operadores morfológicos básicos, Seção 2.4. Os gradientes morfológicos são operadores úteis para aumentar a variedade da intensidade dos *pixels* de uma dada vizinhança por um EE destacando contornos e bordas. A partir da diferença entre a dilatação e a erosão de uma imagem  $f$  usando um EE  $\mathcal{B}$ , denota-se a  $\nabla_{\mathcal{B}}(f)$  como o gradiente morfológico ou *gradiente de Beucher*:

$$\nabla_{\mathcal{B}}(f) = \delta_{\mathcal{B}}(f) - \epsilon_{\mathcal{B}}(f) \quad (2.5)$$

Outro gradiente morfológico pode ser obtido com a diferença de uma imagem  $f$  pela erosão da mesma imagem  $f$  utilizando o EE  $\mathcal{B}$ . Este gradiente morfológico é conhecido como *gradiente interno* ou *meio gradiente por erosão*. Definido como

$$\nabla_{\mathcal{B}}^{-}(f) = f - \epsilon_{\mathcal{B}}(f). \quad (2.6)$$



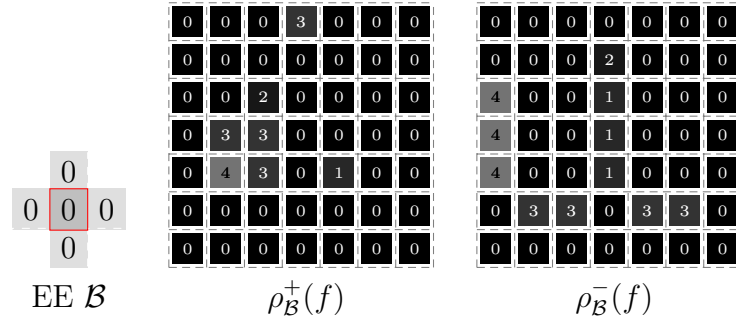
**Figura 2.8:** Exemplo de aplicação de gradiente e gradiente interno com EE formato de cruz em uma dada imagem  $f$ .

Aritmeticamente semelhante ao gradiente interno, os operadores *white top-hat* e *black top-hat* são definidos pela operação entre uma imagem  $f$  e sua abertura ou fechamento. A *white top-hat* de uma imagem  $f$  usando um EE  $\mathcal{B}$  é representada por  $\rho_{\mathcal{B}}^+(f)$  e definida como

$$\rho_{\mathcal{B}}^+(f) = f - \gamma_{\mathcal{B}}(f). \quad (2.7)$$

Ela é idempotente e dual em relação à operação black top-hat. A black top-hat, simbolizada por  $\rho_{\mathcal{B}}^-(f)$ , é obtida pela subtração do fechamento da imagem  $f$  por  $\mathcal{B}$  e  $f$ . Matematicamente, representada como

$$\rho_{\mathcal{B}}^-(f) = \phi_{\mathcal{B}}(f) - f. \quad (2.8)$$



**Figura 2.9:** Exemplo de aplicação de white top-hat e black top-hat com EE formato de cruz em uma dada imagem  $f$ .

## 2.6 OPERADOR DE CONVOLUÇÃO

O operador de convolução é uma transformação linear entre uma imagem e um filtro. O filtro da convolução, também chamado por kernel, é definido como uma função de uma grade retangular para os números reais, ou seja,  $h : \mathbb{Z}^2 \rightarrow \mathbb{R}$  onde  $n_W$  e  $n_H$  são respectivamente a largura e a altura da grade retangular que representa o domínio do

filtro. Assim, a convolução de uma imagem  $f$  aplicada por um filtro  $h$  é definido pela Equação 2.9, isto é:

$$[Conv(f, h)](p) = \sum_{i=-\frac{n_W}{2}}^{\frac{n_W}{2}} \sum_{j=-\frac{n_H}{2}}^{\frac{n_H}{2}} f(p_x + i, p_y + j) \times h(i, j) \quad (2.9)$$

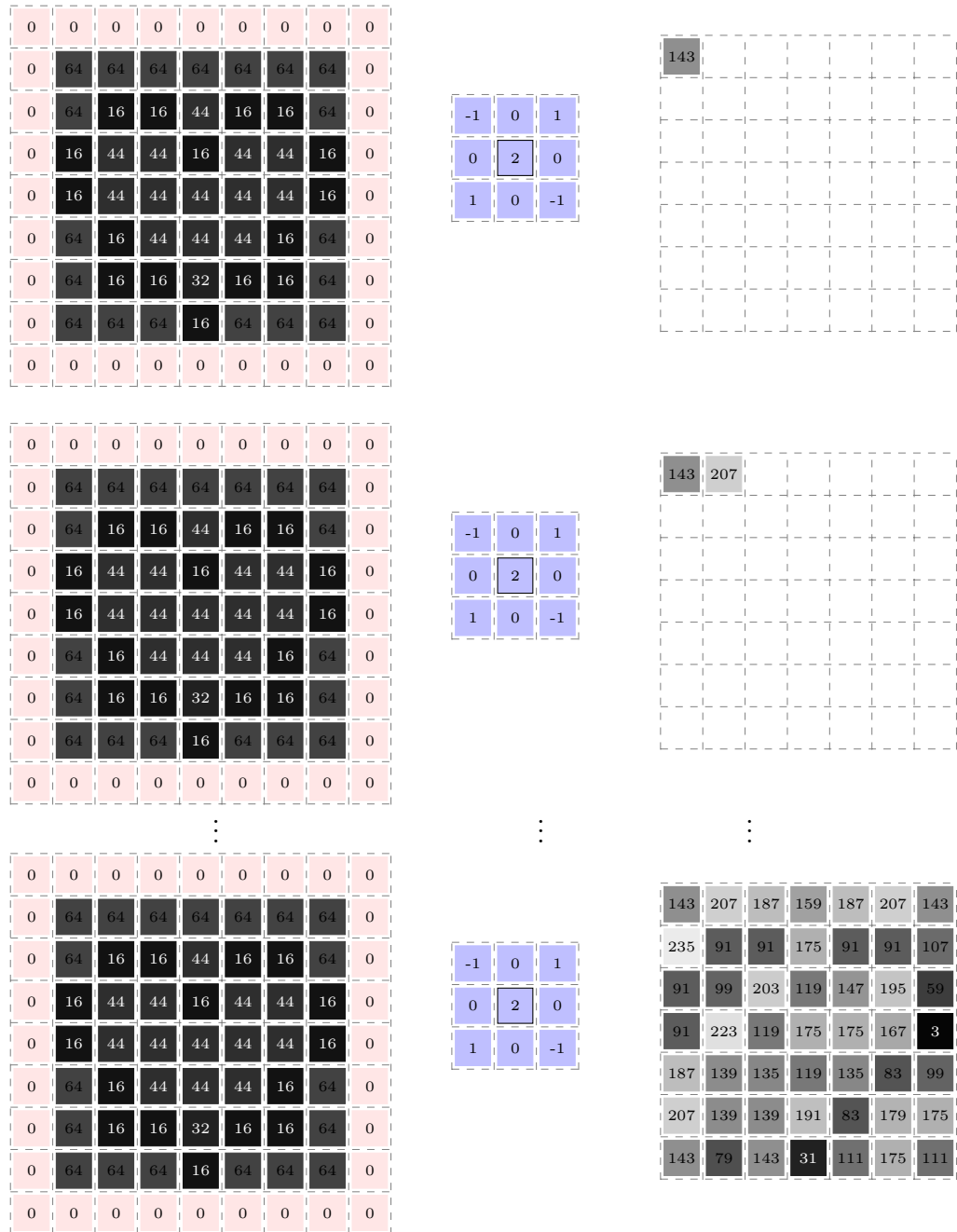
A convolução de uma dada imagem  $f$  ocorre pela sucessiva aplicação do sobreajuste do filtro  $h$  sobre um conjunto de *pixels* da imagem aplicando-se a operação de convolução e deslocamento do filtro pela imagem de acordo com o *stride* definido e é o valor do *stride* que define o número de *pixels* de deslocamento do filtro sobre os *pixels* da imagem para a aplicação da operação.

Para uma melhor aplicação dos filtros convolucionais nas bordas das imagens, é comum a utilização de *padding*, ou adição de *pixels* às bordas da imagem. Então, ao se utilizar *padding*, torna-se possível o posicionamento do centro do filtro convolucional nos *pixels* localizados próximos da borda da imagem. A imagem ou banda resultante de uma convolução terá seus valores relativos aos *padding*, *stride* e dimensões do filtro utilizados na convolução da imagem. Isto é, a altura do *padding*  $p_H$ , a altura do filtro  $h_H$  e o *stride*  $s$ , influenciam na altura da imagem ou banda resultante da convolução bem como a largura do *padding*  $p_W$ , a largura do filtro  $h_W$  e o *stride*  $s$  influenciam na largura da imagem ou banda resultante da mesma convolução. A partir da altura  $n_{H_0}$  e largura  $n_{W_0}$  da banda ou imagem original, pode-se definir a altura  $n_H$  e  $n_W$  da banda ou imagem resultante conforme as Equações 2.10 e 2.11.

$$n_H = \left\lfloor \frac{n_{H_0} + 2p_H - h_H}{s} \right\rfloor + 1 \quad (2.10)$$

$$n_W = \left\lfloor \frac{n_{W_0} + 2p_W - h_W}{s} \right\rfloor + 1 \quad (2.11)$$

A convolução de uma imagem  $f$  utilizando-se um filtro  $h$  pode ser ilustrada conforme a Figura 2.10. Onde, nesta figura, nota-se a imagem original  $f$ , o incremento do *padding* a mesma, e a aplicação da primeira, segunda e última etapa da convolução desta imagem.



**Figura 2.10:** *Convolução etapa a etapa de uma imagem  $f$ : na primeira linha, a representação da imagem  $f$ , a imagem  $f$  com os pixels do padding em rosa, o filtro convolucional em azul com centro destacado e o primeiro valor da imagem convolucionada; na segunda linha repete-se a imagem, a imagem com padding, o filtro utilizado e acrescenta-se o segundo valor ao lado do primeiro valor da imagem convolucionada; na terceira linha, reticências ao processo; e na quarta linha, volta-se a repetir a imagem, a imagem com padding, o filtro utilizado, mas acrescenta-se os valores restantes da convolução.*

---

## CONCEITOS SOBRE APRENDIZAGEM PROFUNDA

---

### Resumo do capítulo

*Este capítulo trata de conceitos básicos que podem ser encontrados em livros que abordam aprendizagem de máquina de forma geral visando simplificar o entendimento do presente trabalho. Os conceitos e definições apresentados neste capítulo foram retirados de livros como *Goodfellow, Bengio e Courville (2016)*, *Norvig e Russell (2014)* e *Haykin (2001)*.*

### 3.1 INTRODUÇÃO À APRENDIZAGEM DE MÁQUINA SUPERVISIONADA

Antes da apresentação de conceitos de *aprendizagem profunda* pretende-se, para melhor entendimento, abordar princípios básicos da *aprendizagem de máquina* como algoritmos de aprendizagem de máquina capazes de aprender tarefas a partir de dados. Uma definição formal de aprendizagem fornecida por *Mitchell (1997)*, diz que: um programa de computador aprende por meio da experiência  $E$  com respeito a alguma classe de tarefa  $T$  e medida de desempenho  $P$ , se seu desempenho nas tarefas  $T$ , medidas por  $P$ , melhoram com a experiência  $E$ .

Nesse sentido, a tarefa não deve ser confundida com o processo de aprendizagem, pois aprender significa atingir a habilidade de executar uma tarefa. Desta forma, tarefas de aprendizagem de máquina são como sistemas devem processar as *entradas*<sup>1</sup>. Dentre as possíveis tarefas desempenhadas por algoritmos de aprendizagem de máquina supervisionada estão a *classificação* e a *regressão*. Em uma tarefa de classificação (por exemplo, classificar se na imagem o paciente tem hipertrofia adenoideana) o computador especifica qual é a classe que uma entrada pertence (no exemplo dado, paciente com hipertrofia ou sem hipertrofia). Enquanto em uma tarefa de regressão (por exemplo, estimar o grau da hipertrofia adenoideana) o computador especifica um valor numérico para uma entrada.

Para tanto, o algoritmo de aprendizagem de máquina (mais precisamente, o algoritmo de treinamento) é geralmente solicitado a produzir uma função  $h$ , conhecida por hipótese, que mapeia o conjunto das entradas  $X$  para o conjunto de saídas esperadas  $Y$ . O conjunto de entradas  $X$  pode ser representado por uma matriz com  $m$  amostras (dimensão das linhas) com  $n$  características (dimensão das colunas) e o conjunto de saídas esperadas  $Y$  pode ser representado por um vetor com  $m$  saídas. Observe-se que esses dados de entradas e saídas esperadas representam as experiências  $E$  da definição do *Mitchell (1997)*.

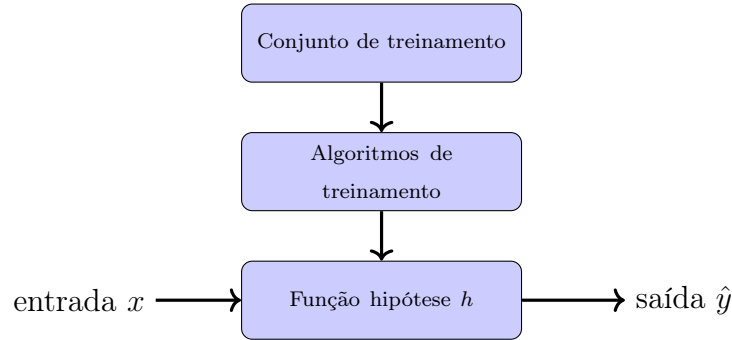
Dadas essas considerações, a grosso modo, um algoritmo de treinamento (como ilustrado na Figura 3.1) toma como entrada um subconjunto de entradas e saídas, denominado

---

<sup>1</sup>Entende-se como entrada um vetor com  $n$  *características* ou medidas quantitativas de algum evento ou objeto que o sistema de aprendizagem de máquina deverá processar.



$X_{\text{train}} \subset X$  e  $Y_{\text{train}} \subset Y$  respectivamente, conhecido por conjunto de treinamento. O conjunto de treinamento é utilizado pelo algoritmo de treinamento para produzir uma função hipótese  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  que tem o melhor desempenho quando comparada aos dados do conjunto de treinamento. Ademais, após o treinamento usa-se a função hipótese  $h$  para prever uma saída  $\hat{y}$  para uma nova entrada  $x \in \mathbb{R}^n$ , ou seja,  $\hat{y} = h(x)$ .



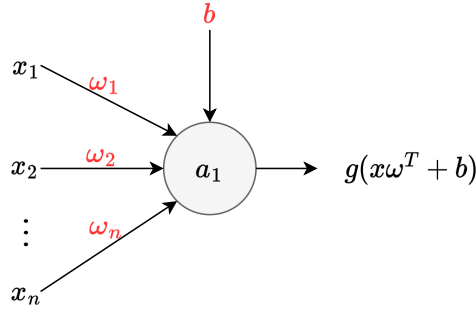
**Figura 3.1:** Processo de geração da função hipótese a partir de um algoritmo de treinamento.

Dentre os algoritmos de treinamento, os algoritmos baseados no gradiente descendente são as mais usadas em aprendizagem profunda. O gradiente descendente é um algoritmo iterativo de otimização utilizado para minimizar uma função de custo que depende de um conjunto de parâmetros que serão otimizados. A função de custo mensura o quão bom são as previsões dadas pela função hipótese  $h$  quando comparadas com as saídas esperadas. Assim, o objetivo é ajustar os valores dos parâmetros da função hipótese  $h$  de forma a minimizar o valor da função de custo. Nas seções seguintes são apresentados os elementos necessários para a compreensão da aprendizagem profunda.

### 3.2 REDES NEURAIS ARTIFICIAIS

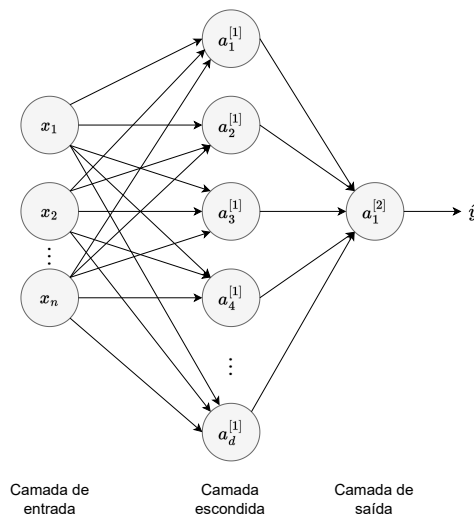
Na década de 1940, [Mcculloch e Pitts \(1943\)](#) relacionavam o funcionamento do cérebro humano com modelos matemáticos, criando o conceito de neurônio artificial, que mapeava o comportamento de um neurônio biológico em um modelo matemático simples. Na década seguinte, [Rosenblatt \(1958\)](#) introduziu o conceito de aprendizagem para o neurônio artificial, desenvolvendo o algoritmo Perceptron. Uma representação visual do neurônio artificial é apresentado na Figura 3.2 onde os elementos  $x_1, x_2, \dots, x_n$  do vetor de entrada  $x$  representam as entradas do neurônio  $a_1$ , os elementos  $b$  e  $\omega_1, \omega_2, \dots, \omega_n$  do vetor  $\omega$  representam os pesos sinápticos do neurônio  $a_1$ , a combinação linear  $x\omega^T + b = \left(\sum_{i=1}^n x_i\omega_i\right) + b$  é o modelo matemático que representa o núcleo do neurônio  $a_1$  e a sua saída é dada por uma função de ativação  $g$  que será definida mais adiante. Esse modelo matemática pode ser visto como uma unidade de processamento que toma como entrada um vetor e devolve como saída um valor escalar dada pela função de ativação  $g$ , isto é  $a_1 = g(x\omega^T + b)$ .

A partir dessa época, pesquisadores começaram a criar redes de neurônios artificiais, o que levou ao desenvolvimento das Redes Neurais Artificiais (RNAs) de múltiplas ca-



**Figura 3.2:** Representação visual de um neurônio artificial do tipo Perceptron.

mas. No entanto, treinar uma RNA de múltiplas camadas era difícil, pois exigia um algoritmo eficiente para ajustar seus pesos sinápticos (parâmetros) das conexões entre os neurônios. Em 1986, [Rumelhart, Hinton e Williams \(1986\)](#) introduziu um algoritmo para essa finalidade, chamado de *backpropagation*, que é usado até hoje para treinar RNAs de múltiplas camadas de Perceptrons (mais precisamente, MLP - *Multilayer Perceptron*). Na Figura 3.3 é apresentado uma MLP com 3 camadas de neurônios onde na primeira camada tem-se os neurônios de entradas representando os elementos  $x_1, x_2, \dots, x_n$  do vetor de entrada  $x$ , na segunda camada, conhecida como camada escondida ou oculta, tem-se os neurônios representado pelas unidades de processamentos  $a_1^{[1]}, a_2^{[1]}, \dots, a_d^{[1]}$  de um vetor de neurônios  $a^{[1]}$ , na terceira camada tem-se o neurônio de saída  $a_1^{[2]}$ . Vale destacar que uma MLP sempre tem uma camada de entrada (representando os dados de entrada), um ou mais camadas escondidas (ocultas) e podendo ter quantidades diferentes de neurônios e, por fim, uma camada de saída. Essa disposição organização dos neurônios em camadas de uma MLP é denominado de arquitetura (ou topologia) da rede.



**Figura 3.3:** Representação visual de uma MLP com uma camada escondida.

Para essa MLP apresentada na Figura 3.3, os pesos sinápticos do neurônio  $a_1^{[1]}$  pode ser representado por um vetor  $W_1^{[1]} = [\omega_1, \omega_2, \dots, \omega_n]$  e bias  $b_1^{[1]}$ . Analogamente, os vetores

$W_2^{[1]}, W_3^{[1]}, \dots, W_d^{[1]}$  e os bias  $b_2^{[1]}, b_3^{[1]}, \dots, b_d^{[1]}$  representam os pesos sinápticos dos neurônios  $a_2^{[1]}, a_3^{[1]}, \dots, a_d^{[1]}$ . Assim, pode-se representar todos os pesos da primeira camada escondida por uma matriz  $W \in \mathbb{R}^{d \times n}$  com  $d$  linhas (dimensionalidade das saídas) por  $n$  colunas (dimensionalidade das entradas) e um vetor de bias  $b^{[1]} \in \mathbb{R}^d$ . Assim, com essa matriz e o vetor de bias  $b^{[1]}$  pode-se calcular os neurônios da primeira camada escondida por  $a^{[1]} = g(W^{[1]}x + b^{[1]})$  resultando no vetor  $a^{[1]} = [a_1^{[1]}, a_2^{[1]}, \dots, a_d^{[1]}]$ . Os neurônios  $a_1^{[1]}, a_2^{[1]}, \dots, a_d^{[1]}$  serão as entradas para a próxima camada e esse processo se repete. Por fim, o neurônio de saída é dado por  $a^{[2]} = g(W^{[2]}a^{[1]} + b^{[2]})$  representando a saída da rede. No caso particular dessa MLP, a função hipótese  $h$  é dada por

$$h(x) = g(W^{[2]}g(W^{[1]}x + b^{[1]}) + b^{[2]}), \quad (3.1)$$

onde os parâmetros (pesos sinápticos)  $W^{[1]}, b^{[1]}, W^{[2]}$  e  $b^{[2]}$  serão otimizados (aprendidos) pelo algoritmo de treinamento de tal forma que a saída da rede tenha o menor erro possível para os dados do conjunto de treinamento. É importante ressaltar que os neurônios de uma MLP podem ter funções de ativação diferentes como é apresentado na Seção 3.2.1.

### 3.2.1 FUNÇÕES DE ATIVAÇÃO

As RNAs do tipo MLPs são como uma coleção de unidades conectadas em camadas cujas propriedades variam conforme as propriedades de seus neurônios. Cada neurônio é como uma unidade de processamento que codifica uma função para a realização do processo de ativação. As funções de ativação das camadas escondidas obrigatoriamente devem ser funções de ativação não lineares. Se não utilizarmos funções não lineares uma rede neural simplesmente funciona como um modelo de regressão linear. Isso vem direto do fato de que: a composição de duas funções lineares é também uma função linear. Como é conhecido que um grande número de problemas só são solucionáveis a partir de uma não linearidade. Dessa forma, é preciso considerarmos funções de ativação não lineares, tais como: ReLU, Leaky ReLU, tangente hiperbólica, Sigmoid e Softmax.

As funções de ativação *ReLU* e *Leaky ReLU* definidas respectivamente pelas Equações 3.2 e 3.3 são utilizadas principalmente em camadas escondidas das MLPs. O nome *ReLU* é o acrônimo das palavras *Rectified Linear Unit* que significam Unidade Linear Retificada, e *Leaky*, vazamento. Este “vazamento” é definido pelo valor  $\alpha \neq 0$ , pois se  $\alpha = 0$ , a função é igual a uma *ReLU* convencional.

$$g(z) = \begin{cases} z & , \text{ se } z > 0 \\ 0 & , \text{ caso contrário} \end{cases} \quad (\text{ReLU}) \quad (3.2)$$

$$g(z) = \begin{cases} z & , \text{ se } z > 0 \\ \alpha z & , \text{ caso contrário} \end{cases} \quad (\text{Leaky ReLU}) \quad (3.3)$$

Definidas respectivamente pelas Equações 3.4, 3.5 e 3.6, as funções de ativação Sigmoid, *Softmax* e tangente hiperbólica são geralmente utilizadas na última camada de uma MLP para uma tarefa de classificação. As funções de ativações sigmoide e tangente hiperbólica são ideais para classificação binária enquanto a função *Softmax* foi projetada para classificação multiclasse.

$$g(z) = \frac{1}{1 + e^{-z}} \quad (\text{Sigmoide}) \quad (3.4)$$

$$g(z_{\text{classeRef}}) = \frac{e^{z_{\text{classeRef}}}}{\sum_{\text{classe}=1}^{\text{numClasses}} e^{z_{\text{classe}}}} \quad (\text{Softmax}) \quad (3.5)$$

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (\text{tangente hiperbólica}) \quad (3.6)$$

Como está descrito nas próximas seções, as derivadas das funções de ativação são essenciais para os ajustes dos pesos sinápticos da MLP durante a etapa de treinamento. Por isso, as derivadas das funções de ativação anteriormente apresentadas são definidas como seguem:

$$g'(z) = \begin{cases} 1 & , \text{ se } z > 0 \\ 0 & , \text{ caso contrário} \end{cases} \quad (\text{Derivada da ReLU}) \quad (3.7)$$

$$g'(z) = \begin{cases} 1 & , \text{ se } z > 0 \\ \alpha & , \text{ caso contrário} \end{cases} \quad (\text{Derivada da Leaky ReLU}) \quad (3.8)$$

$$g'(z) = g(z)(1 - g(z)) \quad (\text{Derivada da Sigmoide}) \quad (3.9)$$

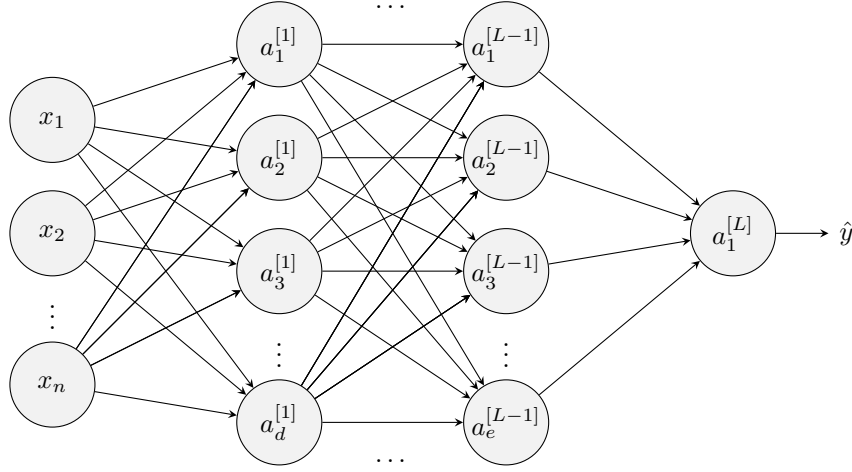
$$g'(z_i) = \begin{cases} g(z_i)(1 - g(z_i)) & , \text{ se } i = j \\ -g(z_i)g(z_j) & , \text{ se } i \neq j \end{cases} \quad (\text{Derivada da Softmax}) \quad (3.10)$$

$$g'(z) = 1 - g(z)^2 \quad (\text{Derivada da tangente hiperbólica}) \quad (3.11)$$

### 3.2.2 PROPAGAÇÃO PARA FRENTE

Em uma RNA do tipo MLP o processo para se obter sua saída é feito a partir do processo de propagação para frente. Esse processo foi executado no final da Seção 3.2 sem

ser previamente declarado. Então, como ilustrado na Figura 3.4, sejam  $x$  um vetor de entradas,  $L$  o número de camadas da MLP (ignorando a camada de entrada),  $W^{[1]}, W^{[2]}, \dots, W^{[L]}$  e  $b^{[1]}, b^{[2]}, \dots, b^{[L]}$  as matrizes e vetores contendo os pesos sinápticos e bias da MLP,  $g^{[1]}, g^{[2]}, \dots, g^{[L]}$  as funções de ativação de cada camada da MLP.



**Figura 3.4:** Modelo abstrato de uma rede neural artificial.

A propagação para frente pode ser definida como a computação sucessiva das unidades de processamento (neurônios) partindo da entrada até atingir sua saída. Mais formalmente

$$\hat{y} = \underbrace{g^{[L]}(W^{[L]} \dots \underbrace{(g^{[2]}(W^{[2]} \underbrace{g^{[1]}(W^{[1]}x + b^{[1]})}_{\text{primeira camada oculta } a^{[1]}} + b^{[2]}) \dots)}_{\text{segunda camada oculta } a^{[2]}} + b^{[L]})}_{\text{camada de saída } a^{[L]}}. \quad (3.12)$$

A camada de entrada é responsável por receber um vetor de entradas  $x = (x_1, \dots, x_n)$  de uma amostra e propagar estes valores para a camada seguinte  $a^{[1]} = g^{[1]}(W^{[1]}x + b^{[1]})$ . É importante notar que  $W^{[1]} \in \mathbb{R}^{d \times n}$  é uma matriz com  $d$  linhas (quantidade de neurônios da segunda camada oculta) por  $n$  colunas (quantidade de entradas) e um vetor de bias  $b^{[1]} \in \mathbb{R}^d$ .

As camadas ocultas são camadas intermediárias, ou seja, camadas localizadas entre as camadas de entrada e saída da MLP. As camadas ocultas são compostas de neurônios que se conectam com todos os neurônios pertencentes a camada anterior ou posterior. Assim, para uma dada camada oculta  $l$ , a matriz  $W^{[l]}$  contém a quantidade de linhas igual a quantidade de neurônios da camada  $l$  e a quantidade de colunas igual a quantidade de neurônios da camada  $l - 1$ . Já o vetor bias  $b^{[l]}$  tem a quantidade de elementos igual a quantidade de neurônios da camada  $l$ .

A última camada da MLP, ou seja a camada de saída, recebe os valores dos neurônios da última camada oculta, processa-os em um ou mais de um neurônio e emite a saída em valor único, para um neurônio, ou em vetor no caso de mais de um neurônio da camada

de saída. A saída da rede é o resultado da função hipótese  $h$  para uma dada entrada  $x$ , ou seja,  $h(x) = \hat{y}$

### 3.2.3 FUNÇÃO DE CUSTO

Para treinar uma RNA é preciso uma forma de mensurar o seu erro (custo) de saída e isso é feito através de uma função de custo. Nesse sentido, o gradiente descendente é um algoritmo de otimização usado para encontrar o mínimo local da função de custo. Assim, dado uma amostra  $(x_i, y_i) \in (X_{\text{train}}, Y_{\text{train}})$  do conjunto de treinamento, a função de custo  $J$  toma como entradas a saída predita pela função hipótese  $\hat{y}_i = h(x_i)$  e a saída esperada  $y_i$  e devolve uma medida de erro entre  $y_i$  e  $\hat{y}_i$  que mensura a diferença entre essas saídas.

Dessa forma, a grosso modo, o algoritmo do gradiente descendente começa com um conjunto inicial de parâmetros  $W^{[1]}, W^{[2]}, \dots, W^{[L]}$  e  $b^{[1]}, b^{[2]}, \dots, b^{[L]}$  para função hipótese  $h$  (ou seja, a rede neural) e, a cada iteração, calcula o gradiente da função de custo  $J$  em relação a esses parâmetros. Em seguida, ele atualiza os parâmetros na direção oposta do gradiente, diminuindo gradualmente a função de custo. Esse procedimento é repetido até a convergência.

A escolha da função de custo é realizada de acordo com a tarefa a ser resolvida. A seguir são apresentados alguns exemplos. Para tanto, dado um conjunto de treinamento  $(X_{\text{train}}, Y_{\text{train}})$ , considere  $\hat{Y} = h(X_{\text{train}})$  como sendo um vetor com as previsões das amostra  $X_{\text{train}}$ .

- Para tarefa de regressão a função de custo Erro Médio Quadrático (MSE) é uma das mais usadas e é definida da seguinte forma:

$$J(\hat{Y}, Y_{\text{train}}) = \frac{1}{|\hat{Y}|} \sum_{i=1}^{|\hat{Y}|} [(\hat{Y}_i - Y_{\text{train}_i})^2], \quad (3.13)$$

e sua derivada é:

$$\frac{\partial J}{\partial \hat{Y}} = [\hat{Y} - Y_{\text{train}}]. \quad (3.14)$$

- Para tarefa de classificação binária a função mais usada é a Entropia Cruzada (EC) definida como a seguir:

$$J(\hat{Y}, Y_{\text{train}}) = -\frac{1}{|\hat{Y}|} \sum_{i=1}^{|\hat{Y}|} [Y_{\text{train}_i} \log(\hat{Y}_i) + (1 - Y_{\text{train}_i}) \log(1 - \hat{Y}_i)], \quad (3.15)$$

e sua derivada é:

$$\frac{\partial J}{\partial \hat{Y}} = -\left[ \frac{Y_{\text{train}}}{\hat{Y}} - \frac{1 - Y_{\text{train}}}{1 - \hat{Y}} \right]. \quad (3.16)$$

onde a derivada  $\frac{\partial J}{\partial \hat{Y}} \in \mathbb{R}^L$  é um vetor com quantidade de elementos igual ao número

de neurônios da camada de saída.

### 3.2.4 RETROPROPAGAÇÃO

O processo de aprendizagem de uma MLP é tradicionalmente realizado por meio de um algoritmo baseado no gradiente descendente. Ele é um algoritmo iterativo de otimização utilizado para minimizar a função de custo  $J$  em função dos seus parâmetros  $W^{[1]}, W^{[2]}, \dots, W^{[L]}$  e  $b^{[1]}, b^{[2]}, \dots, b^{[L]}$ . Em cada iteração do algoritmo é calculado o gradiente da função de custo  $J$  em relação a esses parâmetros e para isso é utilizado o algoritmo do *backpropagation*. A ideia por trás do *backpropagation* é propagar o erro da saída da MLP para as camadas anteriores. Como a saída  $\hat{y}$  é dado pela função hipótese  $h$  que é uma composição de funções de ativação, a regra da cadeia é utilizado para calcular os gradientes. Assim, atualiza-se os parâmetros  $W^{[1]}, W^{[2]}, \dots, W^{[L]}$  e  $b^{[1]}, b^{[2]}, \dots, b^{[L]}$  camada por camada na direção oposta dos gradiente, diminuindo gradualmente a função de custo.

Formalmente, para uma dada camada  $l$  a derivada em respeito a  $W^{[l]}$  e  $b^{[l]}$  é definida como:

$$\frac{\partial J}{\partial W^{[l]}}, \frac{\partial J}{\partial b^{[l]}}. \quad (3.17)$$

Aplicando a regra da cadeia, pode-se computar as derivadas acima da seguinte forma:

$$\begin{aligned} dW^{[l]} &= \frac{\partial J}{\partial W^{[l]}} = \frac{1}{m} dz^{[l]} a^{[l-1]T} \\ db^{[l]} &= \frac{\partial J}{\partial b^{[l]}} = \frac{1}{m} \sum dz^{[l]} \\ da^{[l-1]} &= \frac{\partial J}{\partial a^{[l-1]}} = W^{[l]T} dz^{[l]} \end{aligned} \quad (3.18)$$

onde  $dz^{[l]} = da^{[l]} g'(z^{[l]})$ ,  $z^{[l]}$  é o produto linear  $z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ ,  $a^{[l]} = g(z^{[l]})$  são os valores de saídas dos neurônios da camada  $l$ ,  $dW^{[l]}$  e  $db^{[l]}$  são respectivamente as derivadas em função dos parâmetros  $W^{[l]}$  e  $b^{[l]}$ . Considerando essa regra, pode-se implementar o *backpropagation* para todas as camadas conforme o Algoritmo 14.

**Algoritmo 1: RETROPROPAGAÇÃO**


---

```

1 início
2    $d_{out} = \frac{\partial J}{\partial \hat{Y}}$ 
3    $dz^{[L]} = d_{out} g'(z^{[L]})$ 
4    $dW^{[L]} = \frac{1}{|Y|} dz^{[L]} a^{[L-1]T}$ 
5    $db^{[L]} = \frac{1}{|Y|} dz^{[L]}$ 
6    $da^{[L]} = W^{[L-1]T} dz^{[L]}$ 
7   para cada  $l=L-1, \dots, 1$  faça
8      $dz^{[l]} = da^{[l+1]} g'(z^{[l]})$ 
9      $dW^{[l]} = \frac{1}{|Y|} dz^{[l]} a^{[l-1]T}$ 
10     $db^{[l]} = \frac{1}{|Y|} dz^{[l]}$ 
11     $da^{[l]} = W^{[l-1]T} dz^{[l]}$ 
12  fim
13 fim
14 retorna  $dW, db$ 

```

---

Depois desse procedimento tem-se as derivadas para todas as camadas. De posse dessas derivadas, o gradiente descendente atualiza os parâmetros  $W^{[1]}, W^{[2]}, \dots, W^{[L]}$  e  $b^{[1]}, b^{[2]}, \dots, b^{[L]}$  de forma a minimizar a função de custo. Assim, a cada iteração (ou época) do algoritmo (ver Algoritmo 2), ocorre a atualização dos pesos sinápticos de acordo com uma taxa de aprendizagem  $\alpha$ .

**Algoritmo 2: GRADIENTE DESCENDENTE**


---

```

1 início
2   para cada  $i = 1, \dots, \text{número de épocas}$  faça
3      $\hat{y}, a, z = \text{Propagação para frente}(x, W, b, L)$ 
4      $dW, db = \text{Retropropagação}(\hat{y}, a, z, L)$ 
5     para cada  $l=1, \dots, L$  faça
6        $W^{[l]} = W^{[l]} - \alpha dW^{[l]}$ 
7        $b^{[l]} = b^{[l]} - \alpha db^{[l]}$ 
8     fim
9   fim
10 fim

```

---

**3.3 REDES NEURAIS CONVOLUCIONAIS**

As redes neurais Convolucionais (CNNs, do inglês *Convolutional Neural Networks*) são uma espécie de RNAs projetadas para o processamento de informações em grade. Elas são aplicadas no processamento 2D de imagens ou espectrogramas de áudio, por exemplo. Mas



também são úteis no processamento 1D de sequências e sinais. Essas redes adotam o termo “convolucionais” devido ao uso da operação convolução presente em seu processo de extração de características em processamentos de imagens (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016). Este processo de extração é acomplado ao início de uma MLP conforme a Figura 3.5 em que as CNNs podem ser interpretadas como uma extensão das MLPs.

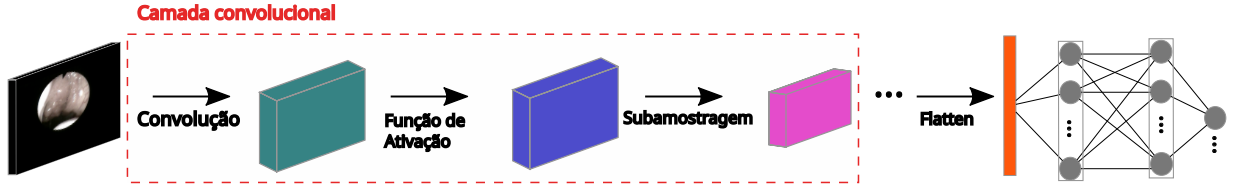
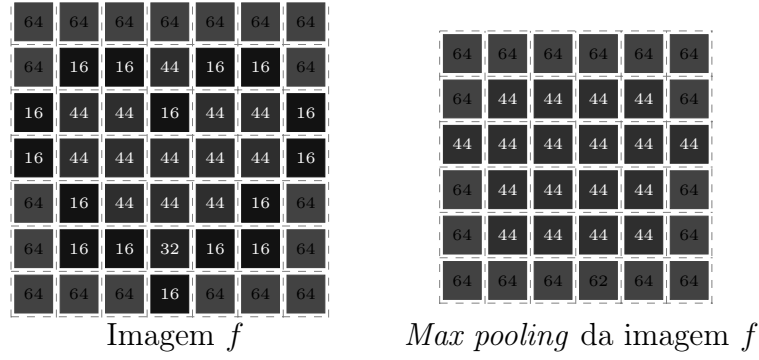


Figura 3.5: Modelo abstrato de uma rede neural convolucional

Ainda na Figura 3.5 é apresentada uma das principais características das CNNs no processamento de imagens: sua entrada e saída. Diferentemente das MLPs tratadas na Seção 3.2 em que suas entradas são um vetor de neurônios  $a^{[l-1]}$  e suas saídas um vetor de neurônios  $a^{[l]}$ , as CNNs possuem um volume  $(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]})$  de entrada e um volume  $(n_H^{[l]}, n_W^{[l]}, n_C^{[l]})$  de saída. Estes volumes são comumente representados por cubos, placas ou caixas em diagramas de CNNs para indicar a presença de uma camada convolucional (KRIZHEVSKY; SUTSKEVER; HINTON, 2017; SILVA et al., 2018; LIU et al., 2021). As camadas convolucionais podem fazer convoluções em um volume de entrada (indicado pela placa predominantemente preta com a imagem de uma adenoide na Figura 3.5), em seguida aplicar uma função de ativação (na placa de tom verde e obter o volume de tom de roxo) e, por fim, fazer uma subamostragem (aplicada ao volume anterior e seu resultado indicado como a placa de cor rosa). É comum a utilização da *ReLU* como função de ativação neste tipo de camada. A subamostragem, que cumpre o papel de reduzir a quantidade de informações de certas regiões dos volumes, pode ser feita a partir de algumas operações como *max pooling* por exemplo (GOODFELLOW; BENGIO; COURVILLE, 2016). Na Equação 3.19 é apresentada a aplicação da operação *max pooling* a partir de um pixel  $p$ .

$$[p_{max}(f)](p) = \max \left\{ \left( p_x - \frac{n_W}{2}, p_y - \frac{n_H}{2} \right), \dots, f \left( p_x + \frac{n_W}{2}, p_y + \frac{n_H}{2} \right) \right\}. \quad (3.19)$$

A operação *max pooling* aplicada a um pixel  $p$  pertencente ao domínio  $\mathcal{D}$  de uma imagem  $f$  é reduzir a informação restrita a uma região  $n_W \times n_H$  a um  $1 \times 1$  utilizando o critério de máximo. Isto é representado de forma gráfica pela Figura 3.6 em que é possível perceber a redução das dimensões da imagem  $f$  de  $7 \times 7$  para  $6 \times 6$  quando o processo é aplicado com um *stride* de valor 1. Processo que consequentemente reduz o número de operações em camadas futuras.



**Figura 3.6:** Exemplo de operação *max pooling* aplicado em regiões  $2 \times 2$  de uma imagem  $f$  utilizando um *stride* de valor 1.

A convolução que ocorre em uma camada convolucional durante a propagação para frente pode ser feita por um ou mais filtros. A profundidade  $n_C^{[L]}$  do volume resultante das convoluções desta camada serão iguais ao número de filtros aplicados. A profundidade do volume será alterada novamente quando passar por uma nova convolução ou camada *flatten*, responsável por converter os volumes em vetores de entrada para camadas MLPs em modelos de CNNs. O processo de propagação para frente das CNNs pode ser interpretado matematicamente como

$$\underbrace{\hat{y} = g^{[L]} \left( \underbrace{\dots flat \left( \underbrace{\dots p_{max}^{[1]} \left( g^{[1]} \left( \underbrace{Conv^{[1]} \left( \underbrace{(f_1, \dots, f_n)}_{\text{Volume de entrada}}, H^{[1]} \right) + b^{[1]} \right)}_{\text{Camada convolucional 1}} \right) \dots \right)}_{\text{Camada de saída L}} \right) \dots + b^{[L]} \right)}_{(3.20)}$$

Na camada convolucional 1,  $H^{[1]}$  representada um vetor de filtros  $h_1, \dots, h_n$  e é um dos parâmetros para a convolução  $Conv^{[1]}$ . A aplicação da convolução  $Conv^{[1]}$  contendo o volume  $(f_1, \dots, f_n)$  e  $H^{[1]}$  como parâmetros pode ser traduzida como um vetor de convoluções  $Conv_1^{[1]}((f_1, \dots, f_n), h_1^{[1]}), \dots, Conv_n^{[1]}((f_1, \dots, f_n), h_n^{[1]})$ . E *flat*, como a operação *flatten*.

A retropropagação do custo de CNNs ocorre de forma semelhante às MLPs. Como parte das CNNs é constituída das MLPs, parte de sua retropropagação é descrita conforme a Seção 3.2.4. A operação *flatten* é ignorada no processo por apenas transformar o volume em vetor. Quanto a subamostragem do exemplo, que utiliza a operação *max pooling*, retropropaga o custo apenas por não possuir filtros para ajuste e sua derivada ser sempre igual a 1. Pois, por exemplo: seja  $f(q_x, q_y) \in \left\{ f\left(p_x - \frac{n_W}{2}, p_y - \frac{n_H}{2}\right), \dots, f\left(p_x + \frac{n_W}{2}, p_y + \frac{n_H}{2}\right) \right\}$  tal que  $f(q_x, q_y) \geq \forall f(p_i, p_j) \in \left\{ f\left(p_x - \frac{n_W}{2}, p_y - \frac{n_H}{2}\right), \dots, f\left(p_x + \frac{n_W}{2}, p_y + \frac{n_H}{2}\right) \right\}$ , então

$$[p_{max}(f)](p) = f(q_x, q_y) \implies \frac{\partial [p_{max}(f)](p)}{\partial f(q_x, q_y)} = 1 (f(q_x, q_y))^{1-1} = 1. \quad (3.21)$$

A retropropagação do custo de CNNs ocorre ao longo de suas camadas convolucionais para atualização dos pesos dos filtros. Este processo é semelhante ao visto na seção 3.2.4 e é aplicado a camadas *poolings* de forma específica. Isto porque as camadas *max pooling*, por exemplo, não possuem pesos em seus filtros. Outro motivo é a derivada da operação ser igual a 1 apenas para o *pixel* de maior intensidade e 0 para o restante. Pois, dada uma adjacência  $\mathcal{A}(p_i)$  qualquer definida pelo filtro *max pooling* em que  $p_j, p_i \in \mathcal{A}(p_i)$  com  $0 < j < (n + 1)$ , a operação será igual a  $f(p_j)$  se somente se  $\forall p \in \mathcal{A}(p_i), f(p_j) \geq f(p)$  e 0, caso contrário. Isto faz com que o processo de retropropagação pela regra da cadeia descrito na seção 3.2.4 apenas repasse o valor de custo para a próxima camada sem ajustar os pesos da camada atual (KRIZHEVSKY; SUTSKEVER; HINTON, 2017; SILVA et al., 2018).

### 3.4 REDES NEURAIS MORFOLÓGICAS

Similar as CNNs descritas na Seção 3.2, as redes neurais morfológicas agregam o termo “morfológicas” pelo uso dos operadores morfológicos, como aqueles apresentados nas Seções 2.4 e 2.5 utilizados em suas camadas (DAVIDSON; HUMMER, 1993). Assim como as CNNs, elas possuem basicamente dois tipos de camadas: as camadas morfológicas e as camadas totalmente conectadas. As camadas morfológicas são compostas por unidades que realizam operações morfológicas, como as apresentadas no Capítulo 2. Essas operações permitem a captura de informações sobre a forma e estrutura dos objetos presente na imagem. As camadas totalmente conectadas, como nas CNNs, possuem a função de combinar as informações extraídas das camadas morfológicas. A partir dessas informações extraídas das imagens são processadas pelas camadas totalmente conectadas em uma saída final que pode ser uma classe (rótulo) ou previsão de uma dada tarefa. De modo geral e abstrato, as redes neurais morfológicas possuem suas camadas dispostas conforme a Figura 3.7.

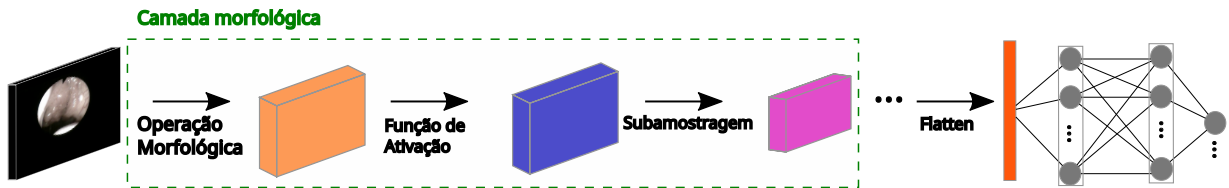


Figura 3.7: Modelo abstrato de uma rede neural morfológica.

Outra semelhança entre as redes neurais morfológicas e as CNNs são as entradas e saídas de suas camadas. Como apresenta a Figura 3.7, as redes neurais morfológicas têm como entrada (placa preta com imagem RGB de uma adenoide no centro) e saída (placa de cor laranja) um volume em sua propagação para frente. A saída produzida pela operação morfológica aplicada ao volume de entrada será a entrada de uma função de ativação. Esta função de ativação também gera um volume de saída (placa de cor roxa) que será o volume de entrada para uma operação de *downsampling*. Operação cujo resultado é um

volume (indicado pela placa de cor rosa) de dimensões verticais e horizontais menores que o volume de entrada. E com a operação *flatten*, o volume de saída da camada morfológica é convertido para um vetor de características como ocorre nas CNNs. Nestes termos, a formalização de uma camada morfológica representada por  $\text{MorphLayer}(\text{Vol}_{\text{in}}, B, g, \rho)$  pode ser escrita como

$$\text{MorphLayer}(\text{Vol}_{\text{in}}, B, g, \rho) = \rho \left( \overbrace{g \left( \underbrace{\text{Morph}(\text{Vol}_{\text{in}}, B)}_{\text{Operação morfológica pelos EEs de B}} \right) + b}_{\text{Função de ativação aplicada em cada elemento}} \right), \quad (3.22)$$

onde a operação  $\text{Morph}(\text{Vol}_{\text{in}}, B)$  é aplicação de uma operação morfológica (isto é, por exemplo: erosão, dilatação, abertura, fechamento, gradiente, entre outros) em cada banda ou *slice* do volume de entrada  $\text{Vol}_{\text{in}}$  por um EE de  $B$ . O volume de saída  $\text{Vol}_{\text{out}} = \text{MorphLayer}(\text{Vol}_{\text{in}}, B, g, \rho)$  terá a profundidade igual ao produto entre profundidade do volume de entrada (número de bandas) pela quantidade de filtros de  $B$  e, as dimensões horizontal e vertical reduzidas pela operação de *downsampling*. A propagação para frente das redes neurais morfológicas pode ser compreendida matematicamente como

$$\hat{y} = \underbrace{g^{[L]} \left( W^{[L]} \dots g^{[1]} \left( W^{[1]} \underbrace{\text{Flatten} \left( \dots \left( \underbrace{\text{MorphLayer}(\text{Vol}_{\text{in}}, B, g, \rho)}_{\text{Operação morfológica pelos EEs de B}} \right) \dots \right) \dots + b^{[1]} \right) \dots + b^{[L]} \right)}_{\text{Camada totalmente conectada 1}}. \quad (3.23)$$

Camada de saída L

A aprendizagem dos pesos sinápticos das redes neurais morfológicas pela utilização do gradiente descendente é semelhante às CNNs. As redes neurais morfológicas utilizam os conceitos descritos na Seção 3.2.4 para fazer a retropropagação do custo pelas suas camadas como é nas CNNs. Esta retropropagação do custo que usa os conceitos da regra da cadeia depende das derivadas das funções da camada morfológica. Em seções anteriores está explicado o processo da propagação do custo pelas operações *flatten*, *max pooling* (como exemplo de *downsampling*) e funções de ativação. Contudo, as operações morfológicas usam conceitos próximos da operação *max pooling*, por exemplo, quando se trata principalmente da dilatação. Esta proximidade é dada pelo uso do critério do máximo que se assemelha a dilatação como

$$\frac{\partial [\delta_{\mathcal{B}}(f)](p)}{\partial \mathcal{B}(q)} = \begin{cases} 1, & \text{se } f(p+q) + \mathcal{B}(q) = \max_{q \in \mathcal{D}_{\mathcal{B}}} \{f(p+q) + \mathcal{B}(q)\} \\ 0, & \text{caso contrário,} \end{cases} \quad (3.24)$$

onde  $f(p+q) + \mathcal{B}(q) = \max_{q \in \mathcal{B}(\mathcal{D})} \{f(p+q) + \mathcal{B}(q)\}$  é, a grosso modo, um mapeamento das coordenadas do domínio  $\mathcal{D}_{\mathcal{B}}$  para atualização dos valores do EE. A atualização desses valores acontece de forma muito semelhante aos filtros das CNNs utilizando a regra

da cadeia. Assim, para a atualização dos valores do EE da operação erosão é utilizada conceitos semelhantes à Equação 3.24. Definida como

$$\frac{\partial [\epsilon_{\mathcal{B}}(f)](p)}{\partial \mathcal{B}(q)} = \begin{cases} 1, & \text{se } f(p+q) - \mathcal{B}(q) = \min_{q \in \mathcal{D}_{\mathcal{B}}} \{f(p+q) - \mathcal{B}(q)\} \\ 0, & \text{caso contrário.} \end{cases} \quad (3.25)$$

Analogamente, as abordagens utilizadas nas Equações 3.24 e 3.25 são aproveitadas nas demais operações. A título de exemplo, para atualização dos valores do EE usado em uma operação de abertura, é definida como a seguinte equação:

$$\frac{\partial [\gamma_{\mathcal{B}}(f)](p)}{\partial \mathcal{B}(q)} = \frac{\partial [\delta_{\mathcal{B}}(\epsilon_{\mathcal{B}}(f))](p)}{\partial \epsilon_{\mathcal{B}}(f)} \frac{\epsilon_{\mathcal{B}}(f)}{\mathcal{B}(q)}. \quad (3.26)$$

Ainda, as propriedades dos operadores como abertura, por exemplo, podem ser preservadas com EEs diferentes quando respeitada a ordem de aplicação dos operadores (erosão e dilatação). Este entendimento se aplica a camadas morfológicas quando respectivamente erosão e dilatação são aplicadas (MONDAL; DEY; CHANDA, 2020).

### 3.5 REGULARIZAÇÃO E OTIMIZAÇÃO

A aprendizagem de máquina que ocorre pelo ajuste dos pesos pode ocasionar durante o treinamento o sobreajuste da função solução sobre os dados de treinamento. Isto significa que o modelo de aprendizagem de máquina desenvolvido foi capaz de “decorar” os dados durante o treinamento e, conseqüentemente, perdeu seu poder de generalização. Então, quando o modelo é exposto a novos dados, dados de teste, por exemplo, sua performance torna-se insatisfatória ou muito abaixo se comparada aos dados de treinamento.

Embora haja outros fatores que contribuam para que um sobreajuste aconteça, tal como a falta de representatividade dos dados de teste no conjunto de dados de treinamento, o sobreajuste pode ser evitado pela *regularização*. Pode-se encontrar na literatura diversos métodos de regularização entre eles destaca-se: a regularização  $L1$  e  $L2$ , *dropout* e bloco residual (HE et al., 2016).

A tarefa de otimização do aprendizado de máquina está geralmente relacionada à função objetivo e suas regras garantirem que a função de custo para o problema tratado seja convexa. Pois, durante o treinamento de aprendizagem de máquina, pode-se confrontar com casos não convexos e, inviáveis de se aplicar o treinamento. Mas, até mesmo os casos convexos, possuem seus desafios.

### 3.6 HIPERPARÂMETROS

Os algoritmos de aprendizagem de máquina possuem parâmetros que definem seu comportamento durante o treinamento. Estes parâmetros são chamados de *hiperparâmetros* que controlam a capacidade dos modelos. No caso das RNAs do tipo MLP, tem-se por exemplo: a quantidade de camadas, a quantidade de neurônios por camada, o tipo das funções de ativação, entre outros. Estes parâmetros, bem como tantos outros relacionados ao problema e o gradiente descendentes, precisam ser estimados e otimizados. Para esse propósito, é geralmente utilizado um outro conjunto de dados de validação que difere do conjunto de treinamento. Diferente do conjunto de treinamento, usado para aprendizagem dos pesos sinápticos, e o conjunto de teste, usado para avaliar a performance do modelo final, o conjunto de validação tem como propósito ajustar os hiperparâmetros relacionado ao modelo. Dividido um conjunto de dados em dois, o primeiro grupo (chamado de treinamento e validação) é destinado ao treinamento de parâmetros e ajuste dos hiperparâmetros enquanto o outro a avaliação final do modelo. O conjunto de validação geralmente corresponderá em torno de 10% a 30% por cento do primeiro grupo e, o restante, será o conjunto de treinamento. Em conjuntos dados pequenos é interessante aproveitar a maior quantidade de dados possível para treinar o modelo. Nesse sentido, as técnicas baseadas em validação cruzadas são as melhores alternativas, como *k-fold*.

---

TRABALHOS RELACIONADOS

---

**Resumo do capítulo**

*Neste capítulo são explorados diversos trabalhos que abordam exames de imagens relacionados à hipertrofia adenoideana. As metodologias e conclusões dos trabalhos selecionados são apresentados em uma visão geral com a análise de suas inter-relações. Por meio da comparação de suas principais semelhanças e diferenças, o capítulo fornece conexões entre os trabalhos de imagens endoscópicas com trabalhos de visão computacional para uma melhor compreensão sobre os temas.*

**4.1 TRABALHOS ENDOSCÓPICOS**

Tomando como desafio o trabalho de otorrinolaringologistas experientes na distinção de lesões de massa da cavidade nasal, [Girdler et al. \(2021\)](#) testam a viabilidade do uso de aprendizagem profunda como um sistema de detecção automática. As redes neurais convolucionais teriam, em sua proposta, o papel de classificar pólipos nasais e papilomas invertidos. Com resultados de especialistas da área, comparar os resultados do modelo de rede utilizado.

A ResNet-152 foi a rede neural convolucional escolhida por eles para fazer a classificação entre: (1) normal; (2) pólipo nasal; e (3) papiloma invertido. E o uso do algoritmo Grad-CAM para a visualização das áreas de maior peso da imagem para o modelo utilizado. Apresentando com o uso do algoritmo um mapa de calor para a verificação da área de interesse na imagem. Ou como explicam os autores, explicar as decisões do algoritmo na classificação pelas regiões consideradas de maior peso ou de interesse para o mesmo.

Os resultados da matriz de confusão dos autores mostram que os especialistas tiveram desempenho superior na classificação das cavidades que se encontravam normais. No entanto, resultados de desempenho competitivamente inferiores na classificação de pólipos nasais e papilomas invertidos em comparação ao modelo. [Girdler et al. \(2021\)](#) justificam que a semelhança de ambos casos implica um grau de dificuldade elevado tanto para especialistas humanos quanto ao modelo rendendo resultados competitivos nestas classificações. O algoritmo conseguiu extrair detalhes interessantes das imagens clínicas e que possuem um bom potencial para detecção e classificação de pólipos nasais e papilomas invertidos. Contudo, um número maior de conjunto de imagens podem melhorar a acurácia dos diagnósticos do modelo.

[Ali et al. \(2021\)](#) aplicam o aprendizado de máquina para avaliação da qualidade e restauração de imagens de vídeos endoscópicos. Pois, artefatos com desfoque de movimento, bolhas, reflexos especulares, entre outros são obstáculos para uma interpretação visual ou

análise automatizada de vídeos de endoscopia. Dada a ampla utilização da endoscopia em diferentes aplicações clínicas, como afirmam Ali et al. (2021), a restauração de *frames* de vídeos corrompidos seria um problema fundamental de imagens clínicas.

Para muitos artefatos como anteriormente enumerados, a principal contribuição do artigo de Ali et al. (2021) está em identificá-los bem como ajustar a restauração de imagens de acordo. E, mais importante, nos casos de *frames* corrompidos com múltiplos artefatos, a sequência de processos estabelecida deve conseguir identificá-los para uma restauração realista evitando a corrupção do *frame*. Sendo, para isto, a escolha do *frame* a ser restaurado algo fundamental.

Os autores, para garantir uma restauração fiel, propuseram a aplicação do método a algumas imagens tidas como parcialmente corrompidas. Assim, estabeleceram para cada *frame*, uma métrica chamada *Quality Score (QS)* baseada em: (1) categoria de artefatos; (2) área; e (3) localização. Atribuindo pesos a cada uma dessas categorias para o cálculo de uma média ponderada, o valor final distingue os *frames* em três classes: *frames* severamente corrompidos ( $QS < 0,5$ ); parcialmente corrompidos ( $0,5 \leq QS \leq 0,95$ ); e *frames* de alta qualidade ( $QS > 0,95$ ).

Com a aplicação do aumento de dados a partir de *flips* (horizontais e verticais), rotação e mudança na escala de cores, Ali et al. (2021) treinaram algumas redes neurais para se fazer o reconhecimento dos artefatos. Para a utilização de caixa delimitadora, Faster R-CNN, RetinaNet e YOLOv3-spp foram utilizadas. Mas, para uma melhor adequação para os formatos irregulares dos artefatos encontrados nas imagens, as classificações por *pixel* foram feitas por FCN8, ResNet-UNet, PSPNet e DeepLabv3+. Todas avaliadas com resultados muito competitivos.

A aplicação do *framework* composto de GANs proposta pelos autores para a restauração das imagens conseguiu, dentre 10 vídeos endoscópicos selecionados aleatoriamente, restaurar 25% dos *frames* dos vídeos. Por outro lado, 40% dos *frames* dos vídeos foram descartados para análise posterior. Em teste de relevância clínica conduzida, os resultados indicaram que a técnica de restauração proposta não introduziu ou removeu informações clinicamente relevantes. Portanto, demonstrando-se bastante promissora.

Em *Contrastive learning-based Adenoid Hypertrophy Grading Network Using Nasoendoscopic Image*, os autores propõem um modelo de rede multi-escala chamado MIB-ANet para classificação dos graus de hipertrofia adenoideana. Este modelo baseado na rede SimCLR de aprendizado contrastivo aliviaria o sobreajuste causado pela falta de imagens nasoendoscópicas com anotações de alta qualidade.

A aplicação do aumento de dados para geração de novas imagens a partir das imagens nasoendoscópicas ocorreu para se fazer o treinamento da rede baseado no aprendizado contrastivo. Juntamente com os dados originais, os autores criaram um *dataset* nomeado SYSU-SZU-EA subdivido em 4 graduações de adenoide. Posteriormente, utilizado para treinar e estabelecer comparações entre o modelo proposto, AlexNet, VGG16, ResNet50



e GoogleNet.

Os autores [Zheng et al. \(2022\)](#) concluem que houve um alívio no problema de sobreajuste do modelo causado pela falta de imagens de nasoendoscopia de adenoide com alta qualidade de anotações utilizando SimCLR para a melhora do desempenho do MIB-ANet. Em comparação, o modelo proposto classificou as 4 graduações de hipertrofia adenoideana melhor que as redes clássicas citadas anteriormente. Obteve-se melhor precisão, *recall*,  $F_1$  score e acurácia nos resultados experimentais devido à capacidade de se explorar dados não anotados pelo pré-treinamento baseado em aprendizado contrastivo, quando as porções de dados de treinamento estão rotuladas.

## 4.2 OUTROS TRABALHOS

Baseado-se em doenças que podem se apresentar na face, [Hu et al. \(2019\)](#) aplicam método multidisciplinar para investigação de fotos de rostos para a identificação de hipertrofia adenoideana. E, a partir da identificação de características analisadas em fotografias de rostos, identificar o tratamento precoce da obstrução respiratória nasal provocada pela hipertrofia adenoideana.

A hipertrofia adenoideana como uma patologia comum em crianças pode afetar o crescimento maxilofacial e dental causando o aspecto de seta ou formato em “V” da arcada dentária superior, fechamento labial desajustado, entre outros. Contudo, o artigo objetiva investigar as mudanças provocadas pela hipertrofia adenoideana contribuindo principalmente com técnicas de visão computacional e um método para diagnóstico de adenoides por imagens de rostos baseando-se em modelos de aprendizagem de máquina desenvolvidos.

Técnicas de mineração de dados foram utilizadas para associar os 68 pontos de referência ou marcações aos rótulos de rostos de hipertrofia adenoideana para se encontrar algumas informações não descobertas. Pois, para elaboração da matriz de correlação usaram-se os coeficientes de correlação de Pearson extraídos das distâncias traçadas entre estes pontos de referência ou marcadores. E para converter o diagnóstico em um problema de regressão linear, agruparam-se as marcações em nove grupos: (1) contorno direito da face; (2) contorno esquerdo da face; (3) sobrancelha direita; (4) sobrancelha esquerda; (5) dorso nasal; (6) base do nariz; (7) olho direito; (8) olho esquerdo; e (9) boca.

A análise das mudanças dos valores dos coeficientes de correlação a partir de um mapa de calor indicaram as distâncias dos lábios, olhos e nariz como pontos característicos que facilitariam o problema de aprendizagem de máquina. Das técnicas de aprendizagem de máquina que o artigo utilizou, obteve-se como melhor especificidade 89,19% e melhor sensibilidade 88,24% respectivamente utilizando KNN e SVM. Concluindo que a acurácia dos modelos deve ser melhorada para o propósito de aplicar um método de baixo custo de auxílio aos diagnósticos médicos e, aprender mais informações sobre rostos de hipertrofia

adenoideana, como os diferentes efeitos provocados pelos diferentes graus de hipertrofia.

Em seu artigo *A deep-learning-based approach for adenoid hypertrophy diagnosis*, [Shen et al. \(2020\)](#) propõem uma solução para classificação de hipertrofia adenoideana sobre uma quantidade de informações limitadas. Pois, conforme afirma o artigo, a obtenção de uma quantidade grande de dados é difícil. E, conseqüentemente, a aprendizagem de máquina para imagens médicas torna-se mais difícil. Esclarecendo que o motivo para a dificuldade de se obter um número grande de imagens médicas está: (a) na necessidade de se requerir autorização aos pacientes; (b) na falta de amostras normais obtidas pela repetição, pois exames de raio-x, como o caso do artigo, são prejudiciais a crianças.

Para se fazer a classificação dos graus de hipertrofia adenoideana promovidas pelo artigo, foram-se utilizadas imagens de raio-x do cavum faríngeo dentre os vários métodos de diagnóstico. A partir dessas imagens, extraem-se medições ou valores absolutos da adenoide e do espaço nasofaríngeo onde a razão entre estes estabelece a porcentagem da obstrução da via aérea superior. Desta forma, para cada intervalo de valor, uma categoria é estabelecida: (a) normal, quando menor ou igual a 60%; (b) moderada, se maior que 60% e menor ou igual a 71%; e (c) severo, caso contrário.

Adotando este método de medição comumente utilizado na China, [Shen et al. \(2020\)](#) utilizam redes neurais artificiais para aprender a demarcar quatro pontos principais nas imagens de raio-x para extração destas medidas. Para haver maior generalização do algoritmo, é proposto termo de regularização chamado de *VerticalLoss*. Um módulo do cosseno do ângulo dos seguimentos de retas traçados a partir da ligação dos pontos demarcados pela rede neural artificial que penaliza predições em que o alinhamento destes pontos violam restrições de verticalidade nas demarcações.

A função de custo *VerticalLoss*, que utiliza a relação de verticalidade entre os pontos demarcados pela rede neural artificial, demonstrou-se bastante eficiente na melhoria das classificações de hipertrofia adenoideana feitas pelo modelo desenvolvido no artigo. Porém, segundo os autores, interessa-se para trabalhos futuros, formas de aproveitamento do modelo desenvolvido com a transformação deste em um modelo mais versátil. Pois, o modelo desenvolvido fora feito sobre um protocolo de imagens que limita a aplicabilidade do modelo a outros campos do conhecimento.

Os autores [Zhao et al. \(2021\)](#), utilizando modelo proposto por [Shen et al. \(2020\)](#), conduziram experimentos para classificar hipertrofia adenoideana a partir de um conjunto de cefalogramas laterais. Com alterações pequenas, tais como módulos *residuais de atenção* (*attention residual modules*) ou função de custo, o modelo de estudo é referenciado como *HeadNet*. E teve melhor desempenho com o uso de módulos residuais de atenção e funções de perda de rotação e de translação.

Concluiu-se a partir dos resultados obtidos no experimento que o modelo *HeadNet* usando módulos residuais de atenção e com funções de custo de rotação e de translação conseguem demarcar os cefalogramas laterais para a classificação de hipertrofia adenoide-

ana como ferramenta suporte ao diagnóstico. No entanto, há limitações observadas pelos autores, tais como as linhas que conectam as demarcações feitas para a mensuração não acompanhar precisamente a anatomia e não conseguir prover uma informação 3D ou direta do problema. E, mesmo assim, o modelo apresentou alta acurácia e estabilidade na detecção de hipertrofia adenoideana.

O modelo de aprendizagem profunda chamado VGG-Lite inspirado no VGG-16 é uma proposta dos autores [Liu et al. \(2021\)](#) para avaliar a hipertrofia adenoideana a partir de exames de cefalogramas laterais. Usando um conjunto de dados de 1023 imagens de raio-x para estabelecer comparativos entre o modelo proposto, a rede neural artificial VGG-16 e especialistas. Também coleta-se dados a respeito do uso do modelo proposto como assistente aos diagnósticos dos especialistas.

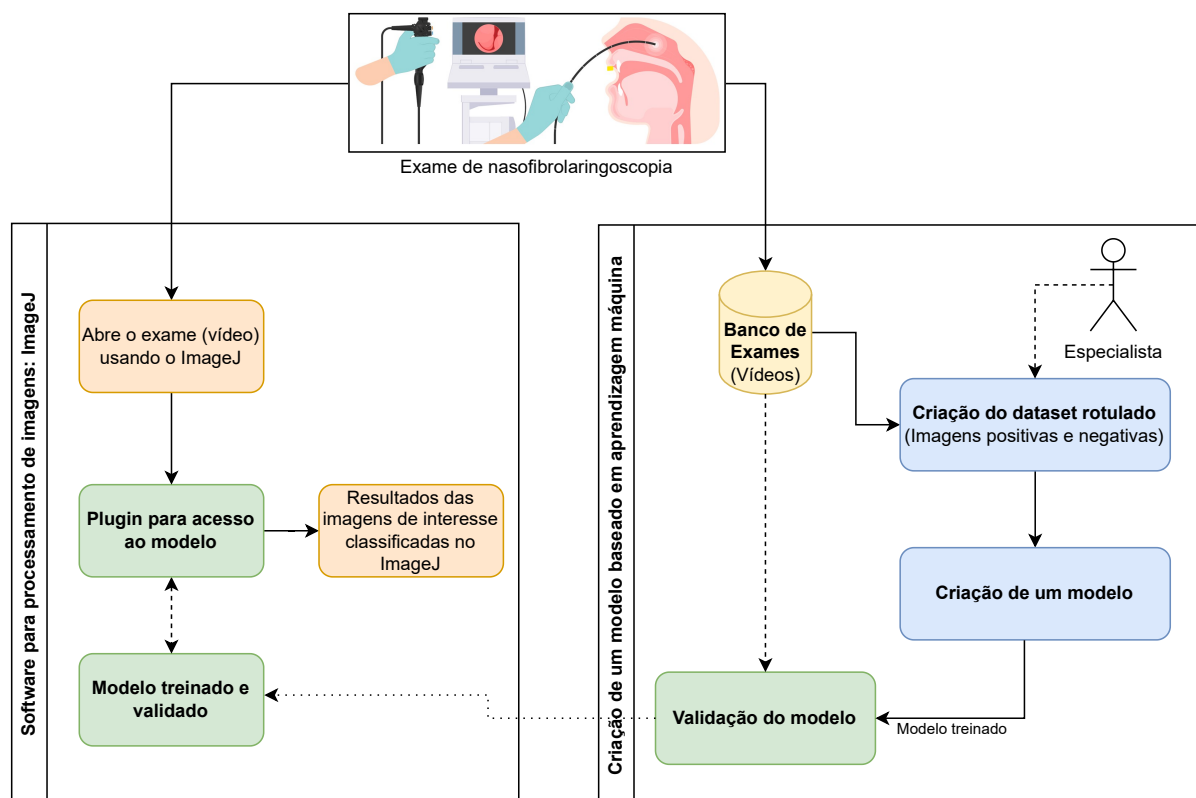
Baseadas na razão entre adenoide e o espaço nasofaríngeo, as classificações foram feitas: (1) para razão inferior a 0.73, adenoide normal; (2) para razão igual ou superior a 0,73, hipertrofia adenoideana. O tempo de diagnóstico do modelo proposto foi significativamente muito inferior comparado ao dos especialistas. Ou seja, o modelo se demonstrou ser mais rápido que eles, inclusive, mais rápido que a rede VGG-16. Entretanto, quando o modelo é assistido por especialistas, têm-se resultados ligeiramente melhores que apenas o modelo ou apenas o especialista. Desta forma, que o modelo pela sua alta acurácia e desempenho apresentou ser uma ferramenta útil nos experimentos para diagnóstico de hipertrofia adenoideana.

## MÉTODO PROPOSTO

**Resumo do capítulo**

*Nesse capítulo é apresentado o método proposto nessa pesquisa no qual consiste em uma ferramenta (software) baseada em aprendizagem de máquina para auxiliar o diagnóstico de hipertrofia adenoideana.*

A ferramenta proposta tem o objetivo detectar a partir do vídeo de um exame de nasofibrolaringoscopia os *frames* (imagens) ideais, ou seja, que são possíveis para médicos especialistas diagnosticar hipertrofias adenoideanas. Para concepção da ferramenta proposta foi considerado as seguintes etapas como pode ser visto na Figura 5.1: (1) Criação de um banco de exames de nasofibrolaringoscopia; (2) Criação de um dataset rotulado por um especialista; (3) Criação de um modelo baseado em redes morfológicas profundas; (4) Criação de um *plugin* para software livre ImageJ para acessar ao modelo proposto treinado e validado. Nas subseções a seguir são apresentados em detalhes cada uma dessa etapas.



**Figura 5.1:** Visão geral do método proposto

### 5.1 CRIAÇÃO DE UMA BANCO DE EXAMES DE NASOFIBROLARINGOSCOPIA

Para criação de um banco de exames de nasofibrolaringoscopia foi estabelecida uma parceria/colaboração com o projeto de pesquisa *Confiabilidade de Nasofibroscopia para avaliação da Hipertrofia Adenoideana e sua correlação com sintomas clínicos* do Departamento de Otorrinolaringologia do Hospital da Irmandade da Santa Casa de Misericórdia de São Paulo. Nesse projeto, após a aprovação por um comitê de ética em pesquisa, foi realizado um estudo com pacientes 20 de ambos os sexos diagnosticados com hipertrofia adenoideana, com idades entre 4 e 14 anos, triados no ambulatório do Departamento de Otorrinolaringologia do Hospital da Irmandade da Santa Casa de Misericórdia de São Paulo no ano 2020, excluíram-se pacientes com: hipertrofia amigdaliana grau III ou IV; malformações craniofaciais; síndrome de down; desordens congênitas; desordens neuromusculares; imunodeficiências; sintomas de infecção de vias aéreas superiores vigentes; desvio de septo nasal obstrutivo.

Mesmo que, sob anestesia tópica em ambas fossas nasais com gel de lidocaína 4% permitindo a gravação da região das cóanas no momento em que a musculatura palatal encontrava-se relaxada para avaliação da obstrução, há o incomodo das crianças para realização do exame. Isso implicou em filmagens mais longas, imagens borradas ou com ruídos que prejudicam a análise e diagnóstico dos especialistas. Nota-se que a decorrência de tais consequências levam o desperdício ou, no melhor dos casos, gasto exagerado de tempo para seleção dos *frames* ou imagens de interesse para realização da análise e diagnóstico dos especialistas.

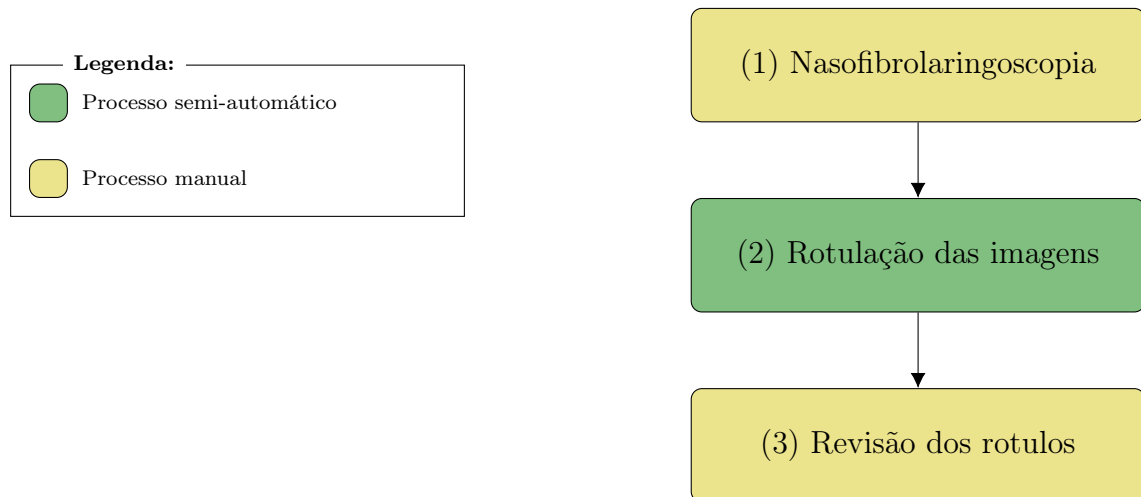
Os vídeos dos 19 exames de nasofibrolaringoscopia contêm duração variando entre 20 segundos a 4 minutos e 13 segundos. Eles foram gravados em sua maioria com uma resolução de  $1080 \times 720$  a uma taxa de 60 *frames* por segundo.

### 5.2 CRIAÇÃO DE UM DATASET ROTULADO

Para a construção de um *dataset* rotulado a partir de vídeos de exames de nasofibrolaringoscopia, elencou-se um conjunto de etapas cuja a execução descrita de forma gráfica está apresentada na Figura 5.2.

A primeira etapa, nasofibrolaringoscopia, consiste nos exames nasofibrolaringoscopia realizados por otorrinolaringologistas. Isto é, em resumo, a introdução do aparelho de nasofibrolaringoscopia pela fossa nasal mais ampla do indivíduo sem ou com a utilização de anestesia tópica permitindo avaliação posterior a partir de gravação de áudio e vídeo.

Após a obtenção dos exames, a segunda etapa, Rotulação das imagens, inicia-se com a análise dos especialistas para definir os intervalos de tempo do exame em que os *frames* positivos (imagens ideais para o diagnóstico de hipertrofia adenoideana) estão. Os *frames* fora do intervalo de tempo indicado pelo especialista são considerados negativos, isto é



**Figura 5.2:** Diagrama de processos da metodologia para rotulação dos frames de interesses dos vídeos dos exames de nasofibrolaringoscopia.

impróprios ou ainda de baixa qualidade para um diagnóstico. A extração de 5 em 5 *frames* dos vídeos foram feitas para fazer a rotulação das amostras em positivas (*frames* positivos) e negativas (*frames* negativos).

Na terceira etapa, Revisão dos rótulos, é dedicada a revisar a segunda etapa, Rotulação das imagens, para aumentar o grau de confiabilidade do dataset. Avaliando os conjuntos de *frames* positivos e negativos, os especialistas indicam quais *frames* estão em desacordo com sua rotulação. Assim, os ajustes na rotulação feitos na segunda etapa são realizados pelos especialistas. É imprescindível que a soma dos *frames* positivos e negativos permaneçam iguais entre as etapas de rotulação das imagens e revisão das imagens, pois nenhum *frame* é descartado. Na Figura 5.3 são apresentados exemplos de *frames* positivos e negativos.

| Classificação | Frames/Imagens |  |  |  |
|---------------|----------------|--|--|--|
| Negativos     |                |  |  |  |
| Positivos     |                |  |  |  |

**Figura 5.3:** Exemplo de frames rotulados como negativo e positivo por especialistas.

A rotulação dos *frames* em positivos e negativos permite agrupá-los por pacientes e tabulá-los de modo a facilitar a visualização de sua distribuição. Esta distribuição é apresentada na Tabela 5.1 em que cada paciente é representado por um identificador (ID do paciente) para a preservação de sua identidade. Deste modo, a distribuição por paciente mostra uma variação grande na quantidade de *frames*. Pacientes que possuem um total de 283 *frames* rotulados a pacientes com 1878 *frames*. Também sendo notória

| ID do paciente | <i>Frames positivos</i> | Porcentagem dos positivos | <i>Frames negativos</i> | Porcentagem dos negativos | Total        | Porcentagem dos frames |
|----------------|-------------------------|---------------------------|-------------------------|---------------------------|--------------|------------------------|
| 01             | 38                      | 6,76%                     | 524                     | 93,24%                    | 562          | 3,02%                  |
| 02             | 17                      | 3,98%                     | 410                     | 96,02%                    | 427          | 2,29%                  |
| 03             | 485                     | 42,40%                    | 659                     | 57,60%                    | 1144         | 6,14%                  |
| 04             | 279                     | 38,22%                    | 451                     | 61,78%                    | 730          | 3,92%                  |
| 05             | 62                      | 8,09%                     | 704                     | 91,91%                    | 766          | 4,11%                  |
| 06             | 191                     | 22,37%                    | 663                     | 77,63%                    | 854          | 4,58%                  |
| 07             | 268                     | 64,89%                    | 145                     | 35,11%                    | 413          | 2,22%                  |
| 08             | 1107                    | 61,40%                    | 696                     | 38,60%                    | 1803         | 9,68%                  |
| 09             | 173                     | 38,96%                    | 271                     | 61,04%                    | 444          | 2,38%                  |
| 10             | 835                     | 51,51%                    | 786                     | 48,49%                    | 1621         | 8,70%                  |
| 11             | 169                     | 59,72%                    | 114                     | 40,28%                    | 283          | 1,52%                  |
| 12             | 530                     | 64,79%                    | 288                     | 35,21%                    | 818          | 4,39%                  |
| 13             | 760                     | 43,38%                    | 992                     | 56,62%                    | 1752         | 9,40%                  |
| 14             | 529                     | 54,93%                    | 434                     | 45,07%                    | 963          | 5,17%                  |
| 15             | 796                     | 53,00%                    | 706                     | 47,00%                    | 1502         | 8,06%                  |
| 16             | 360                     | 42,96%                    | 478                     | 57,04%                    | 838          | 4,50%                  |
| 17             | 365                     | 19,44%                    | 1513                    | 80,56%                    | 1878         | 10,08%                 |
| 18             | 213                     | 29,54%                    | 508                     | 70,46%                    | 721          | 3,87%                  |
| 19             | 729                     | 65,44%                    | 385                     | 34,56%                    | 1114         | 5,98%                  |
| <b>TOTAL</b>   | <b>7906</b>             | <b>42,43%</b>             | <b>10727</b>            | <b>57,57%</b>             | <b>18633</b> | <b>100,00%</b>         |

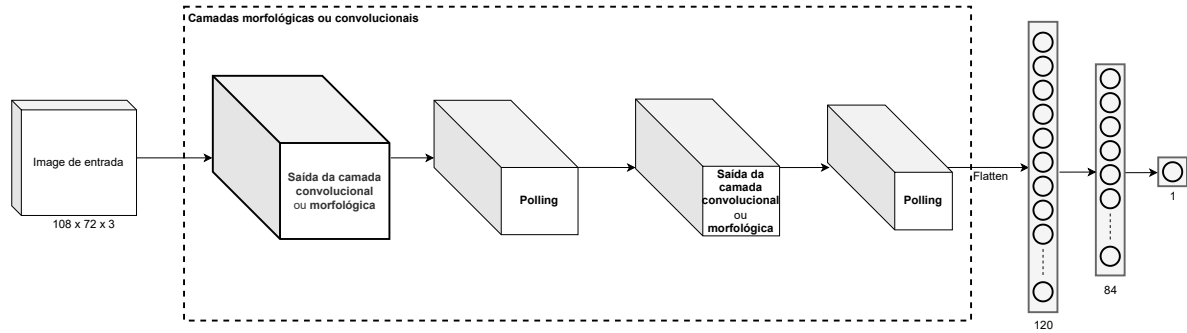
Tabela 5.1: Características do dataset

a alta porcentagem de frames negativos rotulados para cada paciente. O paciente de ID 14, por exemplo, possui todos os seus *frames* rotulado como negativos havendo, portanto, 0,00% de *frames* positivos. E até mesmo no melhor caso cuja a relação é de 70,00% de *frames* positivos e 30,00% negativos, o paciente de ID 06, anteriormente citado, possui apenas um total de 50 *frames*.

### 5.3 CRIAÇÃO DE UM MODELO BASEADO EM REDES MORFOLÓGICAS PROFUNDAS

Para construção do modelo proposto foi utilizado como ponto de partida a arquitetura de redes neurais convolucionais LetNet-5. Desenvolvida por Lecun et al. (1998), a LeNet-5 foi uma das primeiras redes neurais convolucionais a ser utilizada com sucesso. Sua arquitetura é composta por duas camadas de convolução e subamostragens (*pooling*) em sequência, seguidas por camadas totalmente conectadas. O uso da LeNet-5 como base para o modelo proposto tem como finalidade realizar modificações específicas em uma arquitetura já bem estabelecida. Assim, como mostrado na Figura 5.4, almeja-se desenvolver uma arquitetura fundamentada na LeNet-5, incorporando extratores de características mais eficazes. Na construção do modelo, além do operador de convolução utilizada na LeNet-5, são explorados também os operadores morfológicos: dilatação, erosão, abertura, fechamento, gradiente, *white top-hat* e *black top-hat*.

Mais detalhadamente os pontos que são modificados da LetNet padrão incluem: (1) as duas camadas convolucionais; (2) o tamanho e a quantidade de kernels usados pelas

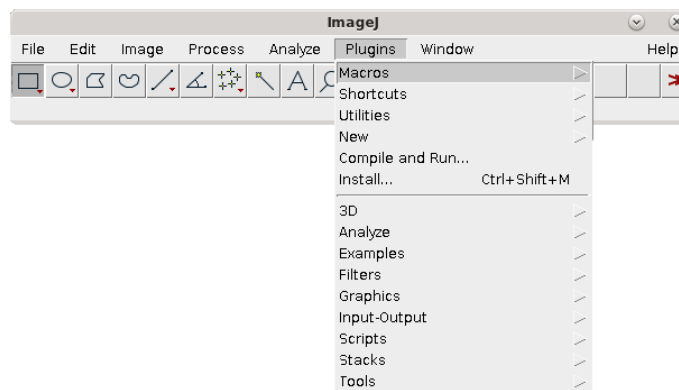


**Figura 5.4:** *Arquitetura proposta da rede neural artificial*

camadas convolucionais. Essas mudanças visam melhorar o modelo para o problema que está sendo resolvido nessa pesquisa. Dessa forma, pretende-se encontrar, utilizando uma busca em grade, a melhor configuração para as camadas de extração de características da rede, ou seja, quais são os melhores dois operadores entre convolução e os morfológicos para a tarefa de detectar *frames* desejáveis. No Capítulo 6 são apresentados resultados da melhor arquitetura para o modelo proposto.

#### 5.4 PLUGIN IMAGEJ

O modelo proposto nessa pesquisa (após treinado, validado e testado) é disponibilizado para o público por meio de plugin para software livre ImageJ. O ImageJ, ver Figura 5.5, é um software de domínio público desenvolvido em linguagem Java que permite editar, processar e analisar imagens e vídeos com disponibilidade para diversos sistemas operacionais como Linux, MacOS e Windows. Ele foi idealizado pelo NIH (National Institute Health) Instituto Nacional de Saúde dos Estados Unidos da América e possui uma arquitetura aberta permitindo ser expandido por meio de *plugins* e macros.



**Figura 5.5:** *Apresentação do menu Plugin do software ImageJ.*

No ImageJ, é possível criar, editar, compilar, invocar e organizar os *plugins* utilizando o menu *Plugins* da janela principal como apresentado na Figura 5.5. Uma das formas de desenvolver *plugins* para o ImageJ é por meio da implementação da interface de progra-



mação *PlugIn* que é apresentada na Figura 5.6.

```
package ij.plugin;

public interface PlugIn {

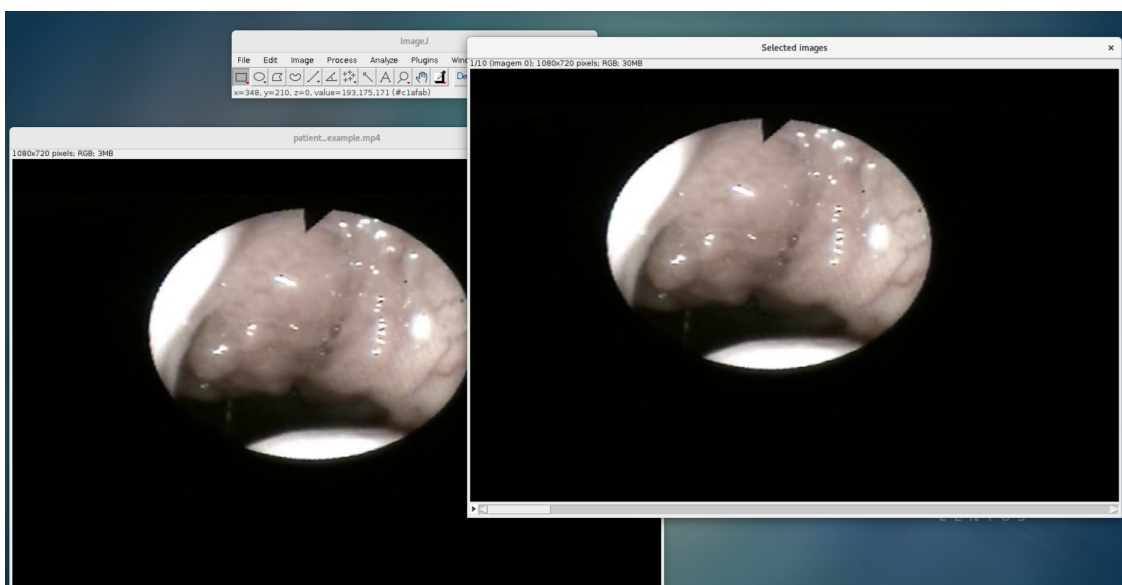
    public void run(String arg);

}
```

**Figura 5.6:** Interface implementação *PlugIn* do *ImageJ*.

Uma vez implementado o *plugin* as instruções de programação implementadas pelo método `public void run(String arg)` são executadas quando o *plugin* é acionado pelo menu do *ImageJ*. Assim, o modelo proposto nessa pesquisa é acessível por meio de um *plugin*.

Para ilustrar o funcionamento da ferramenta para auxiliar no diagnostico de hipertrofia adenoideana proposta nessa pesquisa, considere que o usuário abriu um vídeo de um paciente no qual ele deseja diagnosticar. Com vídeo previamente carregado pelo *ImageJ*, o *plugin* pode ser acionado no menu do *ImageJ*. Dessa forma, o usuário, pode acionar o *plugin* que após sua execução será exibido uma pilha de imagens construídas a partir dos *frames* classificados como positivos pelo modelo proposto. O modelo proposto e utilizado pelo *plugin* classifica cada *frame* do vídeo entre positivos e negativos. Os *frames* que são classificados como positivos e que são ideias para diagnosticar a hipertrofia adenoideana são abertos em uma nova janela como apresentado na Figura 5.7.



**Figura 5.7:** Ilustração do resultado encontrado pelo *plugin* no *ImageJ*

## RESULTADOS

**Resumo do capítulo**

*Este capítulo apresenta os experimentos e resultados realizados nesta pesquisa. Em primeiro lugar, na Seção 6.1, é apresentada a separação das amostras utilizadas no treinamento, validação e teste final do modelo proposto. Em seguida, na Seção 6.2, são discutidos os resultados referentes ao treinamento do modelo, bem como a otimização dos hiperparâmetros, tais como as configurações das camadas de extração de características da rede. Por fim, na Seção 6.3, são apresentados os testes finais realizados no modelo proposto.*

**6.1 DIVISÃO DAS AMOSTRAS PARA TREINAMENTO, VALIDAÇÃO E TESTE**

Conforme apresentado na Seção 5.2, o banco de exames de nasofibrolaringoscopia é composto por 19 amostras, em que cada amostra representa um conjunto de imagens (*frames*) positivos e negativos extraídos do vídeo do exame de nasofibrolaringoscopia realizado por um único paciente. As amostras desse banco de exames são divididas em dois conjuntos: (1) conjunto de treinamento e validação; e (2) conjunto de teste.

O conjunto de treinamento e validação é utilizado por meio de uma validação cruzada para treinar os pesos do modelo e otimizar os hiperparâmetros do modelo, enquanto o conjunto de teste é utilizado para medir a performance final do modelo. É importante destacar que o conjunto de teste é mantido isolado durante todo o processo de treinamento e otimização, a fim de evitar o sobreajuste e garantir que o modelo seja generalizável para novos conjuntos de dados.

| ID do paciente | <i>Frames positivos</i> | Porcentagem dos positivos | <i>Frames negativos</i> | Porcentagem dos negativos | Total         | Porcentagem dos frames |
|----------------|-------------------------|---------------------------|-------------------------|---------------------------|---------------|------------------------|
| 01             | 38                      | 6,76%                     | 524                     | 93,24%                    | 562           | 4,02%                  |
| 03             | 485                     | 42,40%                    | 659                     | 57,60%                    | 1144          | 8,18%                  |
| 04             | 279                     | 38,22%                    | 451                     | 61,78%                    | 730           | 5,22%                  |
| 08             | 1107                    | 61,40%                    | 696                     | 38,60%                    | 1803          | 12,90%                 |
| 10             | 835                     | 51,51%                    | 786                     | 48,49%                    | 1621          | 11,60%                 |
| 11             | 169                     | 59,72%                    | 114                     | 40,28%                    | 283           | 2,02%                  |
| 12             | 530                     | 64,79%                    | 288                     | 35,21%                    | 818           | 5,85%                  |
| 14             | 529                     | 54,93%                    | 434                     | 45,07%                    | 963           | 6,89%                  |
| 15             | 796                     | 53,00%                    | 706                     | 47,00%                    | 1502          | 10,75%                 |
| 16             | 360                     | 42,96%                    | 478                     | 57,04%                    | 838           | 6,00%                  |
| 17             | 365                     | 19,44%                    | 1513                    | 80,56%                    | 1878          | 13,44%                 |
| 18             | 213                     | 29,54%                    | 508                     | 70,46%                    | 721           | 5,16%                  |
| 19             | 729                     | 65,44%                    | 385                     | 34,56%                    | 1114          | 7,97%                  |
| <b>TOTAL</b>   | <b>6435</b>             | <b>46,04%</b>             | <b>7542</b>             | <b>53,96%</b>             | <b>13977%</b> | <b>100,00%</b>         |

**Tabela 6.1:** Amostras do conjunto de treinamento e validação

| ID do paciente | <i>Frames positivos</i> | Porcentagem dos positivos | <i>Frames negativos</i> | Porcentagem dos negativos | Total       | Porcentagem dos frames |
|----------------|-------------------------|---------------------------|-------------------------|---------------------------|-------------|------------------------|
| 02             | 17                      | 3,98%                     | 410                     | 96,02%                    | 427         | 9,17%                  |
| 05             | 62                      | 8,09%                     | 704                     | 91,91%                    | 766         | 16,45%                 |
| 06             | 191                     | 22,37%                    | 663                     | 77,63%                    | 854         | 18,34%                 |
| 07             | 268                     | 64,89%                    | 145                     | 35,11%                    | 413         | 8,87%                  |
| 09             | 173                     | 38,96%                    | 271                     | 61,04%                    | 444         | 9,54%                  |
| 13             | 760                     | 43,38%                    | 992                     | 56,62%                    | 1752        | 37,63%                 |
| <b>TOTAL</b>   | <b>1471</b>             | <b>31,59%</b>             | <b>3185</b>             | <b>68,41%</b>             | <b>4656</b> | <b>100,00%</b>         |

Tabela 6.2: Amostras do conjunto de teste

O *dataset*, ou banco de exames de nasofibrolaringoscopia rotulado, é dividido em porcentagens aproximadas de 70,00% e 30,00% para os respectivos conjuntos de: (1) treinamento e validação; e (2) teste. Pois, sendo o número total de amostras 19, os valores correspondente a 70,00% e 30,00% são respectivamente 13,3 e 5,7. Por se tratar de amostras, estes números foram arredondados para 14 e 5 correspondendo a aproximadamente 73,68% e 26,32%. A seleção de 14 amostras para o conjunto de treinamento e validação ocorreu sem nenhum critério especificado. Isto é, as amostras foram escolhidas de forma arbitrária. Na Tabela 6.1 estão apresentadas as amostras escolhidas para o conjunto treinamento e validação a partir do *dataset* apresentado na Tabela 5.1.

A partir das amostras selecionadas para o conjunto treinamento e validação, a seleção das amostras para o conjunto de teste ocorreu por exclusão. Ou seja, as amostras do *dataset* que não haviam sido escolhidas para compor o conjunto de treinamento e validação são, por conseguinte, amostras do conjunto teste. Estas amostras, bem como algumas informações a respeito dos rótulos de seus *frames*, são apresentados na Tabela 6.2.

## 6.2 SELEÇÃO DE MODELOS E OTIMIZAÇÃO DOS HIPERPARÂMETROS

A linguagem Python em sua versão 3.7 juntamente com TensorFlow versão 2.9 permitiram a utilização do *Morpholayers*, projeto de camadas morfológicas para Keras ou Tensorflow2 com o objetivo de fazer interações entre a morfologia matemática e aprendizagem profunda. Este projeto criado por Velasco-Forero, Pagès e Angulo (2022) facilita a criação de camadas morfológicas para aplicação de operações como a dilatação, erosão, abertura, fechamento, entre outras.

A utilização deste projeto como uma biblioteca é importante na substituição de camadas convolucionais por camadas morfológicas. Isto permite ao modelo LeNet-5 ganhar extratores de características morfológicos como apresenta a Seção 5.3. Dentre os pontos de modificação, tais como a operação da camada, número de filtros, dimensão dos filtros e função de ativação, somente a variação da operação das camadas rende muitos modelos. A substituição das camadas do modelo LeNet-5 foram feitas por camadas morfológicas seguindo os critérios de:

- Dimensão dos filtros: filtros  $3 \times 3$  e filtros  $5 \times 5$ ;
- Quantidade: 3 e 6 filtros para primeira camada e, 12 e 16 filtros para segunda camada;
- Operações: convolução, dilatação, erosão, abertura, fechamento, gradiente, *white top-hat* e *black top-hat* para primeira camada e, convolução, dilatação, erosão, abertura, fechamento e gradiente para segunda camada.

As exclusões das operações *white top-hat* e *black top-hat* para combinações da segunda camada ocorreu pelas limitações da biblioteca. Com estas limitações foram gerados mais de 800 modelos para serem selecionados. As camadas convolucionais utilizadas nas combinações foram preservadas. Deste modo foram considerados 6 filtros  $5 \times 5$  na primeira camada e 16 filtros  $5 \times 5$  na segunda.

A combinação entre duas camadas convolucionais resultará no modelo LeNet-5. Esta combinação foi considerada para estudo da taxa de aprendizagem mais adequada para o treinamento dos demais modelos. A taxa de aprendizagem como um hiperparâmetro fixado favorecerá a comparação entre os demais modelos deixando os impactos das alterações das operações nas camadas mais visíveis. O valor do F-beta *score* com  $\beta = 0,2$  foi utilizado como métrica para a avaliação dos modelos por causa da alta quantidade de amostras de *frames* negativos. Esta alta quantidade de *frames* negativos prejudica o uso da acurácia como uma métrica para avaliar os modelos. Ao classificar todos os *frames* como negativos, o modelo teria ainda uma acurácia significativa. Na Tabela 6.3 são apresentados os valores de F-beta *score* para as variações da taxa de aprendizagem para o modelo LeNet-5.

| Modelo    | Camada 1 | Camada 2 | Taxa de aprendizagem | Fbeta score médio |
|-----------|----------|----------|----------------------|-------------------|
| LeNet-5_0 | Conv     | Conv     | $8 \cdot 10^{-05}$   | 92,42%            |
| LeNet-5_1 | Conv     | Conv     | $4 \cdot 10^{-05}$   | 93,38%            |
| LeNet-5_2 | Conv     | Conv     | $1 \cdot 10^{-05}$   | 91,61%            |

**Tabela 6.3:** F-beta *score* da variação da taxa de aprendizagem para os modelos LeNet-5.

Os melhores parâmetros de treinamentos (por exemplo, taxa de aprendizagem e número de épocas) encontrados para a LeNet-5 (modelo Lenet-5\_1) foram utilizados para o treinamento e seleção das combinações morfológicas. Esta seleção de modelos contém os melhores hiperparâmetros para os modelos cuja primeira camada fixada da rede atende a uma das seguintes operações: convolução (Conv), dilatação (Dil), erosão (Ero), abertura (Open), fechamento (Clos), gradiente (Grad), *white top-hat* (WTHat) e *black top-hat* (BTHat). Estes oito modelos foram selecionados a partir de suas precisões obtidas em *gridsearch* e estão apresentados na Tabela 6.4.

Nesta tabela está descrito respectivamente o nome do modelo, operação da camada 1 (Camada 1), quantidade de filtros da camada 1 (Qtd. de filtros 1), dimensão dos filtros da camada 1 (Dim. dos filtros 1), função de ativação da camada 1 (Função de ativação

1), operação da camada 2 (Camada 2), quantidade de filtros da camada 2 (Qtd. de filtros 2), dimensão dos filtros da camada 2 (Dim. dos filtros 2) e função de ativação da camada 2 (Função de ativação 2).

| Modelo     | Camada 1             | Qtd. de filtros 1 | Dim. dos filtros 1 | Função de ativação 1 | Camada 2   | Qtd. De filtros 2 | Dim. dos filtros 2 | Função de ativação 2 |
|------------|----------------------|-------------------|--------------------|----------------------|------------|-------------------|--------------------|----------------------|
| Conv_Ero   | Convolução           | 6                 | $5 \times 5$       | Tangente Hiperbólica | Erosão     | 12                | $3 \times 3$       | ReLU                 |
| Dil_Conv   | Dilatação            | 6                 | $3 \times 3$       | ReLU                 | Convolução | 16                | $5 \times 5$       | Tangente Hiperbólica |
| Ero_Conv   | Erosão               | 3                 | $5 \times 5$       | ReLU                 | Convolução | 16                | $5 \times 5$       | Tangente Hiperbólica |
| Open_Conv  | Abertura             | 6                 | $5 \times 5$       | Identidade           | Convolução | 16                | $5 \times 5$       | Tangente Hiperbólica |
| Clos_Conv  | Fechamento           | 6                 | $3 \times 3$       | Identidade           | Convolução | 16                | $5 \times 5$       | Tangente Hiperbólica |
| Grad_Conv  | Gradiente            | 6                 | $3 \times 3$       | Identidade           | Convolução | 16                | $5 \times 5$       | Tangente Hiperbólica |
| WTHat_Open | <i>White top-hat</i> | 6                 | $5 \times 5$       | Identidade           | Abertura   | 16                | $5 \times 5$       | Identidade           |
| BTHat_Open | <i>Black top-hat</i> | 3                 | $3 \times 3$       | Identidade           | Abertura   | 16                | $3 \times 3$       | Identidade           |

**Tabela 6.4:** Melhores hiperparâmetros encontrados para um rede baseado na *LetNet-5* onde é fixado na primeira camada uma das seguintes operações: convolução (*Conv*), dilatação (*Dil*), erosão (*Ero*), abertura (*Open*), fechamento (*Clos*), gradiente (*Grad*) e *white top-hat* (*WTHat*) e *black top-hat* (*BTHat*).

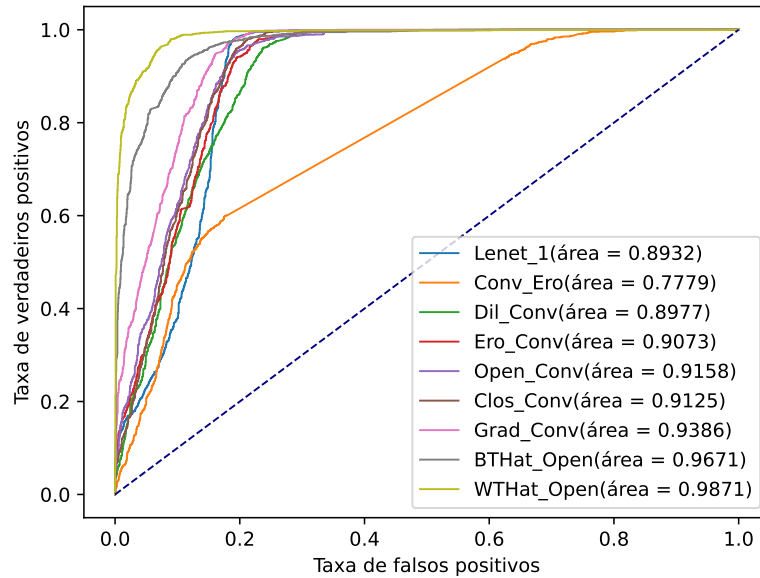
### 6.3 RESULTADOS DOS MODELOS SELECIONADOS

As métricas sobre os dados do conjunto de testes para os oito modelos de selecionados (veja a Tabela 6.4) e o modelo *LeNet-5\_2* (veja a Tabela 6.3) foram obtidas após um novo treinamento feito utilizando o conjunto de treinamento e validação. A partir deste novo treinamento foram feitas predições sobre o conjunto de teste para obtenção das métricas acurácia, precisão, sensibilidade, especificidade,  $F_1$ -score e ROC-AUC (área abaixo da curva ROC) apresentadas na Tabela 6.5.

| Modelo     | Verdadeiros positivos | Verdadeiros negativos | Falsos positivos | Falsos negativos | Acurácia      | Precisão      | Sensibilidade | Especificidade | F1-score      | ROC-AUC       |
|------------|-----------------------|-----------------------|------------------|------------------|---------------|---------------|---------------|----------------|---------------|---------------|
| Conv_Ero   | 822                   | 1882                  | 264              | 759              | 0,7255        | 0,7569        | 0,5199        | 0,8770         | 0,6164        | 0,7779        |
| Dil_Conv   | 1185                  | 1813                  | 333              | 396              | 0,8044        | 0,7806        | 0,7495        | 0,8448         | 0,7648        | 0,8977        |
| Ero_Conv   | 1333                  | 1791                  | 355              | 248              | 0,8382        | 0,7897        | 0,8431        | 0,8346         | 0,8155        | 0,9073        |
| Open_Conv  | 1417                  | 1784                  | 362              | 164              | 0,8589        | 0,7965        | 0,8963        | 0,8313         | 0,8435        | 0,9158        |
| Clos_Conv  | 1448                  | 1764                  | 382              | 133              | 0,8618        | 0,7913        | 0,9159        | 0,8220         | 0,8449        | 0,9125        |
| Grad_Conv  | <b>1519</b>           | 1768                  | 378              | <b>62</b>        | 0,8819        | 0,8007        | <b>0,9608</b> | 0,8239         | 0,8735        | 0,9386        |
| WTHat_Open | 1512                  | 1999                  | 147              | 69               | <b>0,9420</b> | 0,9114        | 0,9564        | 0,9315         | <b>0,9333</b> | <b>0,9871</b> |
| BTHat_Open | 1284                  | <b>2032</b>           | <b>114</b>       | 297              | 0,8897        | <b>0,9185</b> | 0,8121        | <b>0,9469</b>  | 0,8620        | 0,9671        |
| LeNet-5_1  | 1477                  | 1764                  | 382              | 104              | 0,8696        | 0,7945        | 0,9342        | 0,8220         | 0,8587        | 0,8932        |

**Tabela 6.5:** Resultados das métricas de avaliação dos modelos selecionados utilizando o conjunto de teste. Estão destacados em negrito os melhores resultados para cada métrica.

A inclusão do modelo *LeNet-5\_1* na Tabela 6.5 está para estabelecer uma comparação entre um modelo convolucional mais conhecido e os modelos selecionados. O modelo *LeNet-5-2* estabelecerá, portanto, uma base de comparação de desempenho entre os demais modelos. Na Figura 6.1 é apresentado a Curva ROC obtida para os mesmos modelos apresentados na Tabela 6.5. A análise deste resultados são feitas na Seção 6.4 em que são discutidos os valores desta seção.



**Figura 6.1:** Curvas ROC dos nove modelos avaliados.

#### 6.4 ANÁLISE DOS RESULTADOS

De forma geral, os modelos de melhor hiperparâmetros e o LeNet-5\_1, presentes na Tabela 6.5, apresentaram bons resultados em acurácia, precisão, sensibilidade e especificidade. Contudo, modelos mais complexos como Grad\_Conv, BTHat\_Open e WTHat\_Open obtiveram resultados ainda melhores em comparação a outros modelos mais simples compostos basicamente dos operadores morfológicos elementares (dilatação, erosão, abertura e fechamento). Destes modelos mais simples, apenas o modelo Clos\_Open obteve resultados próximos ao LeNet-5\_1. Com resultados melhores que o LeNet-5\_1, os modelos mais complexos com resultados em destaque. Contrapondo-se aos modelos mais simples que tiveram melhores resultados com uma segunda camada convolucional, os três melhores modelos não apresentaram um padrão bem definido. Apenas o BTHat\_Open e WTHat\_Open tiveram sua segunda camada definida como abertura. Tanto a quantidade de filtros quanto suas dimensões não apresentaram nenhuma homogeneidade ou padrões claros. O mais comum à maioria destes modelos foi suas segundas camadas possuírem um total de 16 filtros de dimensões  $5 \times 5$ .

O gráfico ROC dos modelos apresentado na Figura 6.1 ilustra parte dos dados da Tabela 6.5. O gráfico também sugere a alta viabilidade dos modelos mais complexos como bons classificadores ao problema de classificação dos frames ideias para diagnóstico de hipertrofia adenoideana. A linha definida para o modelo LeNet-5\_1 possibilita uma base de comparação em relação ao desempenho classificatório dos demais modelos. As camadas convolucionais mantiveram-se preponderantes no gráfico, porém os melhores modelos foram compostos sem ela.

CONCLUSÃO

---

Neste trabalho, foi apresentado um método baseado em aprendizagem de máquina para auxiliar o diagnóstico de hipertrofia adenoideana que permite detectar frames (imagens) de interesse a partir de um vídeo de exame de nasofibrolaringoscopia. Para a construção dessa ferramenta, foi utilizado um banco de 19 exames de nasofibrolaringoscopia em vídeo. Cada exame teve seus frames rotulados em positivos e negativos para a classificação das imagens de interesse. Após a revisão dos rótulos aplicados as imagens, o banco de exames foi dividido de forma arbitrária em conjuntos de treinamento e validação e, teste. Com base no modelo conhecido como LeNet-5, foram gerados mais de 800 modelos a partir das variações das operações morfológicas em suas camadas. Destes modelos, foram selecionados os oito melhores modelos para novo treinamento com os conjuntos de treinamento e validação e, testados com o conjunto de teste. A partir do conjunto de teste foram obtidas métricas que apontaram o modelo LeNet-5\_1 como o quarto melhor modelo perdendo para Grad\_Conv, BTHat\_Open e WTHat\_Open. Dentre os modelos morfológicos, WTHat\_Open obteve os melhores resultados com 6 filtros  $5 \times 5$  com operador *white top-hat* em sua primeira camada e 16 filtros  $5 \times 5$  com operador abertura em sua segunda. Este modelo para auxiliar no diagnóstico de hipertrofia adenoideana, através de um *plugin* para o software livre ImageJ, é disponibilizado em <https://github.com/GuilhermeBuenoMartins/morph-adenoid-plugin>. Para trabalhos futuros, sugere-se aumentar a base de dados, classificar os graus de hipertrofia adenoideana e a segmentação de imagens à adenoide nos exames de nasofibrolaringoscopia.



## REFERÊNCIAS BIBLIOGRÁFICAS

- ALI, S.; ZHOU, F.; BAILEY, A.; BRADEN, B.; EAST, J. E.; LU, X.; RITTSCHER, J. **A deep learning framework for quality assessment and restoration in video endoscopy**. *Medical image analysis*, Elsevier, v. 68, p. 101900, 2021. Citado na pág. 37, 38.
- BALDASSARI, C. M.; CHOI, S. **Assessing adenoid hypertrophy in children: X-ray or nasal endoscopy?: Assessing Adenoid Hypertrophy in Children**. *The Laryngoscope*, v. 124, n. 7, p. 1509–1510, 2014. ISSN 0023-852X. Citado na pág. 11.
- DAVIDSON, J. L.; HUMMER, F. **Morphology neural networks: An introduction with applications**. *Circuits, Systems and Signal Processing*, Springer, v. 12, n. 2, p. 177–210, 1993. Citado na pág. 33.
- DOUGHERTY, E. R.; LOTUFO, R. A. *Hands-on morphological image processing*. [S.l.]: SPIE press, 2003. v. 59. Citado na pág. 14.
- ESTEVA, A.; CHOU, K.; YEUNG, S.; NAIK, N.; MADANI, A.; MOTTAGHI, A.; LIU, Y.; TOPOL, E.; DEAN, J.; SOCHER, R. **Deep learning-enabled medical computer vision**. *NPJ digital medicine*, Nature Publishing Group, England, v. 4, n. 1, p. 5–5, 2021. ISSN 2398-6352. Citado na pág. 10.
- FRANCHI, G.; FEHRI, A.; YAO, A. **Deep morphological networks**. *Pattern Recognition*, Elsevier, v. 102, p. 107246, jun. 2020. Disponível em: <<https://hal.archives-ouvertes.fr/hal-02922299>>. Citado na pág. 10.
- GIRDLER, B.; MOON, H.; BAE, M. R.; RYU, S. S.; BAE, J.; YU, M. S. **Feasibility of a deep learning-based algorithm for automated detection and classification of nasal polyps and inverted papillomas on nasal endoscopic images**. In: WILEY ONLINE LIBRARY. *International Forum of Allergy & Rhinology*. [S.l.], 2021. v. 11, n. 12, p. 1637–1646. Citado na pág. 10, 37.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado na pág. 10, 12, 14, 22, 31.
- HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2001. Citado na pág. 22.
- 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 (CVPR)*. [S.l.: s.n.], 2016. Citado na pág. 35.
- HU, X.; ZHANG, Q.-Y.; YANG, J.-J.; WANG, Q.; LEI, Y.; WU, J.-L. Photographic analysis and machine learning for diagnostic prediction of adenoid hypertrophy. In: IEEE. *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*. [S.l.], 2019. p. 7–11. Citado na pág. 39.
- KIRSZENBERG, A.; TOCHON, G.; PUYBAREAU, É.; ANGULO, J. **Going beyond p-convolutions to learn grayscale morphological operators**. In: SPRINGER. *Discrete Geometry and Mathematical Morphology: First International Joint Conference, DGMM 2021, Uppsala, Sweden, May 24–27, 2021, Proceedings*. [S.l.], 2021. p. 470–482. Citado na pág. 10.



KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017. Citado na pág. 31, 33.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. *Nature (London)*, Nature Publishing Group, England, v. 521, n. 7553, p. 436–444, 2015. ISSN 0028-0836. Citado na pág. 10, 31.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, n. 11, p. 2278–2324, 1998. Citado na pág. 45.

LIU, J.; LI, S.; CAI, Y.; LAN, D.; LU, Y.; LIAO, W.; YING, S.; ZHAO, Z. **Automated radiographic evaluation of adenoid hypertrophy based on VGG-Lite**. *Journal of Dental Research*, SAGE Publications Sage CA: Los Angeles, CA, v. 100, n. 12, p. 1337–1343, 2021. Citado na pág. 31, 41.

MAAHS, M. A. P.; ALMEIDA, S. T. de. *Respiração oral e apneia obstrutiva do sono: Integração no diagnóstico e tratamento*. [S.l.]: Thieme Revinter Publicações LTDA, 2017. Citado na pág. 11.

MASCI, J.; ANGULO, J.; SCHMIDHUBER, J. **A learning framework for morphological operators using counter-harmonic mean**. In: SPRINGER. *Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, May 27-29, 2013. Proceedings 11*. [S.l.], 2013. p. 329–340. Citado na pág. 10.

MCCULLOCH, W.; PITTS, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 127–147, 1943. Citado na pág. 23.

MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-hill New York, 1997. v. 1. Citado na pág. 22.

MONDAL, R.; DEY, M. S.; CHANDA, B. **Image restoration by learning morphological opening-closing network**. *Mathematical Morphology-Theory and Applications*, De Gruyter Open Access, v. 4, n. 1, p. 87–107, 2020. Citado na pág. 10, 35.

NOGUEIRA, K.; CHANUSSOT, J.; MURA, M. D.; SCHWARTZ, W. R.; SANTOS, J. A. d. **An Introduction to Deep Morphological Networks**. *arXiv preprint arXiv:1906.01751*, 2019. Citado na pág. 10.

NORVIG, P.; RUSSELL, S. *Inteligência artificial: Tradução da 3a Edição*. Elsevier Brasil, 2014. ISBN 9788535251418. Disponível em: <<https://books.google.com.br/books?id=BsNeAwAAQBAJ>>. Citado na pág. 22.

PICCIALLI, F.; SOMMA, V. D.; GIAMPAOLO, F.; CUOMO, S.; FORTINO, G. **A survey on deep learning in medicine: Why, how and when?** *Information Fusion*, v. 66, p. 111–137, 2021. ISSN 1566-2535. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1566253520303651>>. Citado na pág. 10, 12.

- PIGNATARI, S. S. N.; ANSELMO-LIMA, W. terezinha. **Tratado de Otorrinolaringologia**. 3. ed. Grupo Gen, 2017. ISBN 9788595154247. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788595154247/>>. Citado na pág. 11.
- PILTCHER, O. B.; COSTA, S. S. da; MAAHS, G. S.; KUHL, G. **Rotinas em otorrinolaringologia**. [S.l.]: Artmed Editora, 2014. Citado na pág. 11.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958. ISSN 0033-295X. Disponível em: <<http://dx.doi.org/10.1037/h0042519>>. Citado na pág. 23.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. Citado na pág. 24.
- SERRA, J.; SOILLE, P. **Mathematical morphology and its applications to image processing**. 1. ed. [S.l.]: Springer Netherlands, 1994. v. 2. (Computational Imaging and Vision 2, v. 2). Citado na pág. 10.
- SHEN, Y.; LI, X.; LIANG, X.; XU, H.; LI, C.; YU, Y.; QIU, B. **A deep-learning-based approach for adenoid hypertrophy diagnosis**. *Medical Physics*, Wiley Online Library, v. 47, n. 5, p. 2171–2181, 2020. Citado na pág. 10, 11, 40.
- SILVA, F. Machado da; LENZ, M. L.; FREITAS, P. H. C.; SANTOS, S. Cerqueira Bispo dos. **Inteligência Artificial**. Grupo A, 2018. ISBN 9788595029392. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788595029392/>>. Citado na pág. 31, 33.
- SOILLE, P. et al. **Morphological image analysis: principles and applications**. [S.l.]: Springer, 1999. v. 2. Citado na pág. 14.
- VELASCO-FORERO, S.; PAGÈS, R.; ANGULO, J. Learnable empirical mode decomposition based on mathematical morphology. *SIAM Journal on Imaging Sciences*, v. 15, n. 1, p. 23–44, 2022. Citado na pág. 49.
- YSUNZA, A.; PAMPLONA, M. C.; ORTEGA, J. M.; PRADO, H. **Video fluoroscopy for evaluating adenoid hypertrophy in children**. *International journal of pediatric otorhinolaryngology*, Elsevier, v. 72, n. 8, p. 1159–1165, 2008. Citado na pág. 11.
- ZHANG, D.; LU, G. **Review of shape representation and description techniques**. *Pattern recognition*, Elsevier, v. 37, n. 1, p. 1–19, 2004. Citado na pág. 10.
- ZHAO, T.; ZHOU, J.; YAN, J.; CAO, L.; CAO, Y.; HUA, F.; HE, H. Automated adenoid hypertrophy assessment with lateral cephalometry in children based on artificial intelligence. *Diagnostics*, MDPI, v. 11, n. 8, p. 1386, 2021. Citado na pág. 40.
- ZHENG, S.; LI, X.; BI, M.; WANG, Y.; LIU, H.; FENG, X.; FAN, Y.; SHEN, L. **Contrastive learning-based Adenoid Hypertrophy Grading Network Using Nasoendoscopic Image**. In: IEEE. *2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)*. [S.l.], 2022. p. 377–382. Citado na pág. 10, 39.