

**Marcos Vinicius Araujo Almeida**

**Deep Learning para exames  
médicos de imagem**

**PROJETO FINAL**

**DEPARTAMENTO DE INFORMÁTICA**  
Programa de Graduação em Engenharia da  
Computação

Rio de Janeiro  
Novembro de 2023



**Marcos Vinicius Araujo Almeida**

## **Deep Learning para exames médicos de imagem**

### **Relatório de Projeto Final II**

Relatório de Projeto Final, apresentado ao Programa de Engenharia da Computação, do Departamento de Informática da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Augusto César Espíndola Baffa

Rio de Janeiro  
Novembro de 2023

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

**Marcos Vinicius Araujo Almeida**

Graduando em Engenharia da Computação na PUC-Rio.

Ficha Catalográfica

Araujo Almeida, Marcos Vinicius

Deep Learning para exames médicos de imagem / Marcos Vinicius Araujo Almeida; orientador: Augusto César Espíndola Baffa. – 2023.

48 f: il. color. ; 30 cm

Projeto Final - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Trabalho de Conclusão de Curso. 2. Aprendizado Profundo. 3. Redes Neurais de Convolução. 4. Imagens de Exames médicos. 5. Pneumonia. 6. InceptionV3. 7. ResNet50.  
I. Baffa, Augusto. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para a minha mãe e minha madrinha, pelo seu suporte e encorajamento.

## **Agradecimentos**

Para meu orientador Augusto Baffa, por conduzir por todos os caminhos e me ajudar a montar essa tese.

Para minha mãe, que sempre deu todo seu suor e esforço para fornecer a melhor educação possível para mim.

Para minha madrinha, meia-irmã e todas as outras pessoas que participaram da minha criação, contribuindo para o que eu sou hoje.

Agradeço a PUC-Rio pela oportunidade de ser bolsista integral.

Agradeço a todos os amigos que conheci durante esses 5 anos de faculdade pelo suporte e apoio.

## Resumo

Araujo Almeida, Marcos Vinicius; Baffa, Augusto. **Deep Learning para exames médicos de imagem**. Rio de Janeiro, 2023. 48p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

As Redes Neurais Convolucionais (CNNs) representam um avanço significativo no campo de imagens médicas, redefinindo os métodos de diagnóstico e análise. Sua eficácia se destaca na identificação e classificação de anormalidades, facilitando a detecção precoce de condições como câncer, lesões cerebrais e problemas cardíacos. Nessa tese, foram comparados modelos baseados em CNNs para determinar o mais eficiente na tarefa de classificar imagens de raio-X pulmonares, com o objetivo de diagnosticar a presença ou ausência de pneumonia. Este trabalho destacou o potencial das CNNs em aplicações práticas, evidenciando sua relevância e eficácia no diagnóstico médico por imagem.

## Palavras-chave

Aprendizado Profundo; Redes Neurais de Convolução; Imagens de Exames médicos; Pneumonia; InceptionV3; ResNet50;.

## **Abstract**

Araujo Almeida, Marcos Vinicius; Baffa, Augusto (Advisor). **Deep Learning for Medical Imaging Examinations**. Rio de Janeiro, 2023. 48p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Convolutional Neural Networks (CNNs) represent a significant advancement in the field of medical imaging, redefining diagnostic and analytical methods. Their effectiveness is particularly notable in the identification and classification of abnormalities, promoting the early detection of conditions such as cancer, brain injuries, and cardiac issues. In this thesis, CNN-based models were compared to decide the most efficient one for the task of classifying pulmonary X-ray images, aimed at diagnosing the presence or absence of pneumonia. This work highlighted the potential of CNNs in practical applications, underscoring their relevance and efficacy in image-based medical diagnosis.

## **Keywords**

Deep Learning; Convolution Neural Networks; Medical Imaging; Pneumonia; InceptionV3; ResNet50.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
<b>2</b>	<b>Situação Atual</b>	<b>15</b>
<b>3</b>	<b>Objetivos</b>	<b>17</b>
<b>4</b>	<b>Pesquisas Realizadas</b>	<b>19</b>
4.1	Pneumonia	19
4.2	Redes Neurais	21
4.3	Convolutional Neural Networks	24
4.4	ResNet50	26
4.5	InceptionV3	27
4.6	Otimizador RMSprop	29
4.7	Otimizador Adam	30
<b>5</b>	<b>Projeto e Especificação do Sistema</b>	<b>33</b>
5.1	Bibliotecas	33
5.2	Análise do Dataset	35
<b>6</b>	<b>Implementação e Avaliação</b>	<b>36</b>
6.1	Análise e Processamento dos Dados	36
6.2	Criação dos Modelos	39
6.3	Resultados	39
<b>7</b>	<b>Considerações Finais</b>	<b>43</b>
<b>8</b>	<b>Referências bibliográficas</b>	<b>45</b>



## Lista de figuras

Figura 3.1	Exemplos de imagem de entrada	17
(a)	Pulmão afetado por pneumonia	17
(b)	Pulmão saudável	17
Figura 3.2	Exemplo de resultado do modelo	18
Figura 4.1	Pneumonia afetando um pulmão	19
Figura 4.2	Incidência de pneumonia em adultos	20
Figura 4.3	Neurônio Matemático	22
Figura 4.4	Funções de ativação	23
(a)	Sigmóide	23
(b)	ReLU	23
(c)	Tangente Hiperbólica	23
Figura 4.5	Exemplo de stride	25
Figura 4.6	Exemplo de padding	25
Figura 4.7	Visão completa de uma CNN	25
Figura 4.8	Arquitetura da ResNet50	27
Figura 4.9	Arquitetura da InceptionV3	29
Figura 4.10	Performance entre otimizadores	32
Figura 6.1	Raio-X de pulmões	37
Figura 6.2	Dados dos estágios	38
Figura 6.3	Comparação de acurácia e perda	41

## Lista de tabelas

Tabela 6.1	Resultados do treinamento	40
Tabela 6.2	CNN - Matriz de confusão	41
Tabela 6.3	ResNet50 - Matriz de confusão	42
Tabela 6.4	InceptionV3 - Matriz de confusão	42

## Lista de algoritmos

Algoritmo 1	Processamento dos dados	36
Algoritmo 2	Criação e treinamento de um modelo	39

**Lista de Códigos**

Código 1	Bibliotecas utilizadas	33
----------	------------------------	----

## **Lista de Abreviaturas**

CNN – *Convolutional Neural Networks*

IA – Inteligência Artificial

NLP – *Natural language processing*

CAM – *Class Activation Map*

ADAM – *Adaptive Moment Estimation*

RMSProp – *Root Mean Square Propagation*

RGB – *Red, Green & Blue*

ReLU – *Rectified Linear Unit*

SGD – *Stochastic Gradient Descent*

*Nah, I'd win.*

**Satoru Gojo**, *Jujutsu Kaisen*.

# 1

## Introdução

A cada dia que passa percebemos cada vez mais os impactos da tecnologia no nosso cotidiano, sendo eles positivos ou negativos. Desde inteligências artificiais, que podem nos dar respostas à perguntas simples (HAQUE, 2023), economizando nosso tempo de pesquisa, até *cyber-ataques* que podem contribuir para o vazamento de dados sensíveis de milhões de pessoas (FANDI et al., 2023).

A Inteligência Artificial, uma área emergente e cada vez mais robusta na tecnologia, simula o conhecimento humano através de máquinas, destacando-se na otimização de processos e no incremento da produtividade. Deep Learning é uma subárea do aprendizado de máquina (*Machine Learning*), que por sua vez é uma subárea da IA. Ela é capaz realizar tarefas relativamente mais complexas como reconhecimento e análise de imagens, reconhecimento de voz e linguagem natural. De acordo com o estudo da PwC foi previsto que até 2030 a Inteligência Artificial pode contribuir em até \$15.7 trilhões para a economia global.(PWC, 2017)

Seguindo por essa lógica, muito entusiastas realizaram diversos projetos para aplicar essas soluções inovadoras na área da saúde. Segundo a empresa de consultoria americana, *Accenture*, até 2026, os Estados Unidos irão economizar US\$ 150 bilhões por ano com cuidados com a saúde por consequência da aplicações de IA.(FILHO, 2018) Dito isso, será mostrado nesse relatório, um exemplo de aplicação da Inteligência artificial na área da saúde e possíveis melhoras para criação e enriquecimento de modelos de aprendizado de máquina.

## 2

### Situação Atual

A inteligência artificial na medicina consiste no uso de algoritmos de aprendizado de máquina, aliado com grandes bases de dados, para promover análises mais precisas de exames e melhorar a experiência dos pacientes. Muitos desses modelos de aprendizado fornecem suporte para várias partes do mundo, tanto dentro de hospitais ajudando médicos a tomarem melhores decisões, como também em pesquisas de novos fármacos.

Primeiramente, podemos perceber uma frente bem relevante na área de desenvolvimento de novos medicamentos. O processo de criação até teste que é realizado hoje é extremamente custoso, podendo levar anos de pesquisa. Segundo o artigo feito por Mathew Chun (CHUN, 2023), com a chegada de novos algoritmos de redes neurais, como NLP, diversas áreas desse ramo foram afetadas e estão contribuindo positivamente para o aperfeiçoamento de novos remédios. Melhoras significativas foram notadas na aquisição de melhores alvos de genes para servirem de contrariantes para determinadas doenças; Predição de propriedades dos fármacos antes mesmo da testagem; Geração de padrões moleculares de remédios, nunca antes vistos.

O uso da IA para detecção de doenças e produção de diagnósticos mais precisos se tornou um aliado para diferentes clínicos, contribuindo para o aumento de sua produtividade. Justamente pelo fato de máquinas não terem as mesmas limitações físicas que um humano normal, uma IA pode coletar dados de pacientes como, batimentos cardíacos, respirações por minuto, informações sobre a pressão, enquanto o médico não estiver presente e notificar quando notar algo fora dos padrões. Como dito no estudo da IBM, um cliente da empresa conseguiu desenvolver um modelo preditivo de inteligência artificial para detectar condições de sepse grave em bebês pré-maturos. (IBM, Acessado em 23/06/2023)

A inteligência artificial também pode ser aplicada no exame de imagens médicas. Esse ramo é muito importante na medicina, já que elas são necessárias para a visualização de órgãos internos com o objetivo de detectar anomalias em sua estrutura ou funcionamento. Muitos dispositivos podem ser utilizados, gerando imagens ou vídeos de qualidade, sendo eles, raios-x, tomografias computadorizadas, ressonâncias magnéticas, tomografias por emissão de pósitrons (PET), e ultrassonografias. Após a captura, essas imagens devem ser analisadas e compreendidas para a detecção precisa de anormalidades. Caso seja identificada alguma inconformidade, devem ser apresentadas sua localização exata,



tamanho e forma da anomalia. Hoje, sistemas de saúde mais inteligentes, tem como objetivo aplicar IA para realizar essas tarefas de análise, que antes eram feitas por médicos treinados com sua experiência adquirida ao longo dos anos.

Estudos indicam que o uso de IA's em imagens médicas podem ser tão eficiente quanto um radiologista detectando sinais de câncer de mama em um exame. De acordo com o centro de tecnologia de Massachusetts, foi criado um modelo de aprendizado de máquina capaz de detectar o crescimento de um tumor maligno na mama de uma paciente, 5 anos antes de se formar completamente e deixá-la em condições graves. (CONNER-SIMONS, 2019) O modelo foi treinado com dados de mais de 60.000 pacientes e, o mesmo conseguiu detectar padrões sutis no tecido mamário que indicariam futuros casos de tumores na região.

### 3

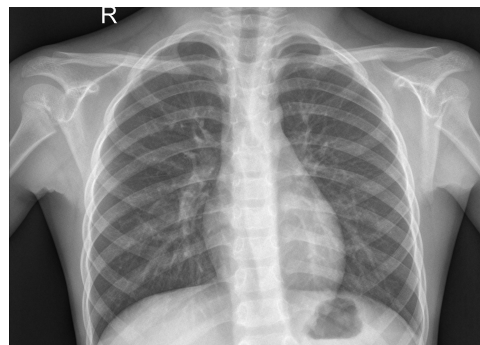
## Objetivos

Como dito no capítulo anterior, o ramo de imagens médicas é extramente importante para a medicina e aplicar técnicas de inteligência artificial para automatizar o processo de análise, pode ser um ganho extremamente positivo. Portanto, o propósito desta tese é explorar diferentes modelos de inteligência artificial que possam classificar, com base em imagens de raio-X do pulmão, se um indivíduo possui pneumonia. O estudo visa comparar os resultados de diferentes modelos e variações para identificar o mais eficaz entre as opções disponíveis. Para tal, serão empregadas diversas métricas e testes na avaliação desses modelos.

Em primeiro momento, serão testados diversos tipos de imagens, incluindo exemplos de pulmões saudáveis e outros afetados por pneumonia, como ilustrado nas figuras 3.1a e 3.1b.



(a) Pulmão afetado por pneumonia



(b) Pulmão saudável

Figura 3.1: Exemplos de imagem de entrada

Após recebidas essas entradas, os modelos devem, de alguma maneira, classificar as imagens em **NORMAL** ou **PNEUMONIA**, retornando sua confiança para aquela previsão. Um exemplo de saída pode observado na imagem 3.2.

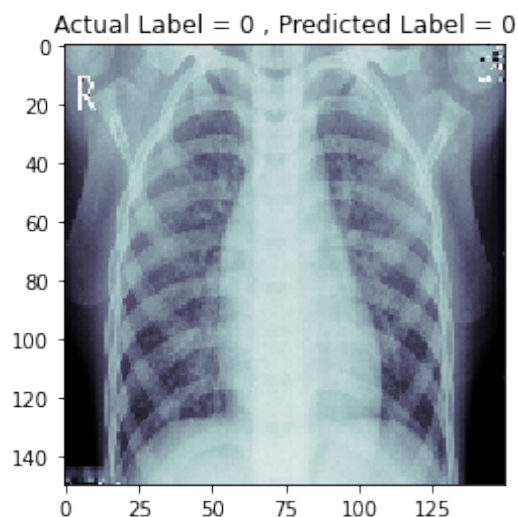


Figura 3.2: Exemplo de resultado do modelo

Os dados obtidos vêm, por padrão, com um gabarito anexado, o que possibilitou a comparação direta. Contudo, na aplicação prática, o resultado se limitaria à classe prevista pelo modelo.

## 4

### Pesquisas Realizadas

Nesse capítulo serão abordadas todas as pesquisas realizadas para a confecção dessa tese. Os tópicos variam desde temas mais técnicos explorando diferentes tipos de redes neurais e otimizadores, até assuntos mais voltados para a área médica.

#### 4.1

##### Pneumonia

A pneumonia é uma infecção que inflama os sacos de ar em um ou ambos os pulmões, onde esses sacos podem se encher de líquido ou pus, provocando tosse com pus ou muco, febre, calafrios e dificuldade para respirar, podendo ser melhor representado na imagem 4.1. A condição pode ser causada por uma variedade de organismos, incluindo bactérias, vírus e fungos. A gravidade da pneumonia pode variar de leve a grave, sendo mais perigosa para bebês e crianças pequenas, pessoas com mais de 65 anos e pessoas com problemas de saúde ou sistemas imunológicos enfraquecidos (World Health Organization, 2023). Podemos confirmar esses detalhes também na imagem 4.2, onde observamos que em todas as condições as pessoas mais afetadas por pneumonia são idosas. Essa imagem retrata um estudo feito da incidência de casos de pneumonia nos EUA dos anos de 2007 até 2010 (FROES; BLASI, 2017).

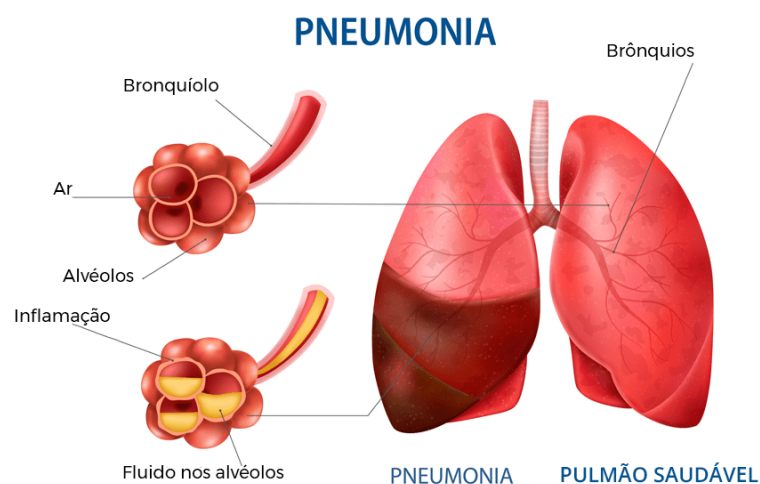


Figura 4.1: Pneumonia afetando um pulmão

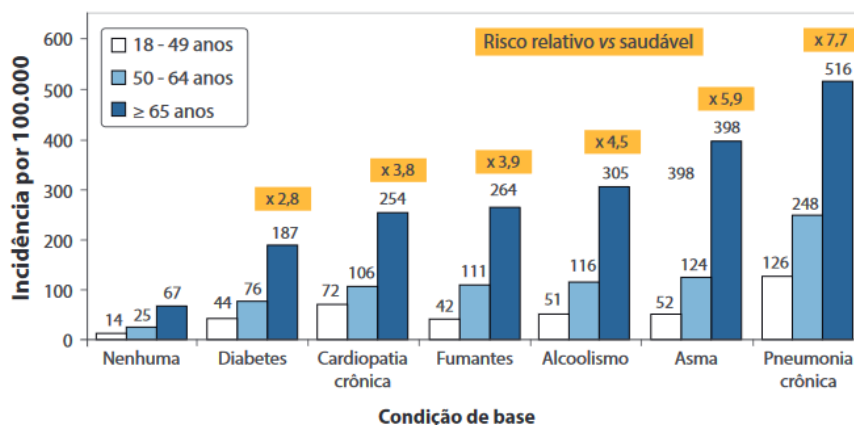


Figura 4.2: Incidência de pneumonia em adultos

No Brasil, o Sistema Único de Saúde (SUS) registra anualmente mais de 600 mil internações por Pneumonia Adquirida na Comunidade (PAC) e Influenza. De janeiro a agosto de 2022, houve 44.523 mortes por pneumonia no país, um aumento significativo em relação ao mesmo período do ano anterior, que teve 31.027 óbitos. (BRASIL, 2023)

Em um contexto global, em 2019, a pneumonia foi responsável pela morte de cerca de 2,5 milhões de pessoas. Notavelmente, quase um terço dessas vítimas eram crianças com menos de 5 anos, fazendo da pneumonia a principal causa de morte para crianças nessa faixa etária. Apesar de ainda muitas crianças morrerem devido à pneumonia hoje, desde 1990 houve uma redução de mais de três vezes nas taxas de mortalidade infantil devido à doença em todo o mundo. (DADONAITE; ROSER, 2019) Em 2019, a pneumonia resultou na morte de 740.180 crianças com menos de cinco anos, correspondendo a 14% das mortes totais nessa faixa etária e a 22% das mortes entre crianças de um a cinco anos, impactando famílias globalmente (World Health Organization, 2023).

A pneumonia bacteriana, que é a mais comum, é frequentemente causada pela bactéria *Streptococcus pneumoniae*, embora outras bactérias como *Mycoplasma pneumoniae* e *Chlamydophila pneumoniae* também possam ser responsáveis. A pneumonia viral, por outro lado, é uma causa comum de pneumonia em crianças, com vírus respiratórios sendo os culpados frequentes. A pneumonia fúngica pode ocorrer em pessoas com sistemas imunológicos enfraquecidos (World Health Organization, 2023).

Os sintomas da pneumonia podem variar, mas geralmente incluem tosse produtiva, febre, calafrios, fadiga, sudorese, dor no peito, confusão (em pessoas

idosas) e falta de ar. O diagnóstico é geralmente feito com base nos sintomas e confirmado através de um exame físico, radiografia de tórax e, possivelmente, exames de sangue e culturas de escarro (World Health Organization, 2023).

O tratamento para pneumonia depende da causa. A pneumonia bacteriana é geralmente tratada com antibióticos, enquanto a pneumonia viral pode ser tratada com antivirais. É também importante descansar, manter-se hidratado e tomar medicamentos para aliviar os sintomas. Para prevenção, existem vacinas como a vacina pneumocócica e a vacina contra a gripe, e medidas como lavar as mãos regularmente e evitar contato com pessoas doentes podem ser eficazes (World Health Organization, 2023).

As complicações da pneumonia podem ser graves, podendo incluir infecção bacteriana no sangue (sepse), acúmulo de líquido e pus no espaço ao redor do pulmão (empema), e insuficiência respiratória. Indivíduos com sistemas imunológicos enfraquecidos, doenças crônicas, ou que estão em certos grupos etários (como crianças muito pequenas ou idosos) estão em maior risco de desenvolver pneumonia (CONTROL; PREVENTION, 2022).

## 4.2

### Redes Neurais

Redes neurais é uma abordagem computacional ao processamento de informação, similar ao sistema de nervoso dos seres humanos, especialmente o cérebro. Ela é uma sub-área de aprendizado de máquina, chamado de aprendizado profundo (*deep learning*) capaz de realizar operações mais complexas e que necessitem de um poder computacional maior.

Sua unidade fundamental é chamada de neurônio, sendo apenas uma simplificação matemática do neurônio biológico.

Como ilustrado na figura 4.3, num primeiro momento, as informações são recebidas pelo neurônio em termos matemáticos (representados por  $x_1, x_2, \dots, x_n$ ), ou seja, deve representar um valor numérico. Isso acontece porque, em uma rede neural, neurônios são conectados uns aos outros de maneira encadeada, por isso, é interessante manter esse padrão. Cada entrada de informação é associada a um peso. Esses pesos são valores numéricos ajustáveis (indicados na imagem por  $w_{k1}, w_{k2}, \dots, w_{kn}$  e representam a importância ou influência que uma determinada entrada pode impactar na saída do neurônio. Vale ressaltar que esses pesos inicialmente são definidos aleatoriamente e logo após ajustados durante o treinamento da rede (GOODFELLOW; BENGIO; COURVILLE, 2016), (WATT; BORHANI; KATSAGGELOS, 2016).

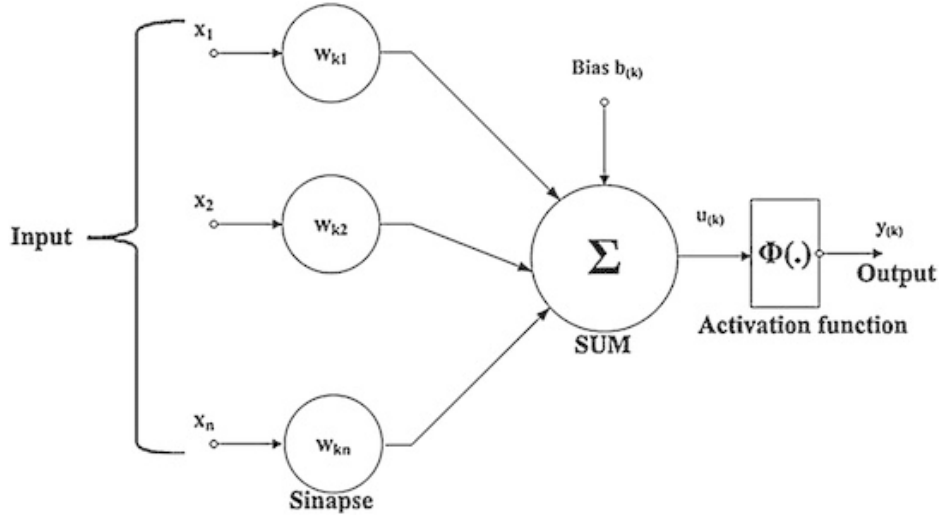


Figura 4.3: Neurônio Matemático

Recebendo essas entradas, o neurônio multiplica cada uma pelo seu respectivo peso e, em seguida, soma todos os produtos encontrados. Isso é chamado de função de soma, sendo representada na imagem pela letra grega "Σ". Normalmente esse processo de soma é representado matematicamente por um produto escalar, de um vetor de entradas  $X$  por um vetor de pesos  $W$ .

$$Soma = w_1 \times x_1 + w_2 \times x_2 + \dots + w_n \times x_n$$

Uma passo importante antes de calcular a soma ponderada é adicionar uma constante chamada de Viés (*Bias*, representado como  $b_{(k)}$  na imagem). O viés permite ajustar a saída do neurônio junto com os pesos, proporcionando um grau de liberdade adicional à rede neural. Podemos entender ele como uma maneira de ajustar a "sensibilidade" do neurônio. Após essa "correção" do viés, na soma ponderada, seu resultado é passado por uma função de ativação  $\Phi(.)$ , que introduz uma não-linearidade, permitindo que a rede neural modele operações mais complexas. As funções mais utilizadas são: **Sigmóide** (4.4a), **ReLU** (4.4b), **Tangente Hiperbólica** (4.4c), **Softmax** (sendo frequentemente usada na camada de saída de problemas de classificação) e entre outros (GOODFELLOW; BENGIO; COURVILLE, 2016), (WATT; BORHANI; KATSAGGELOS, 2016).

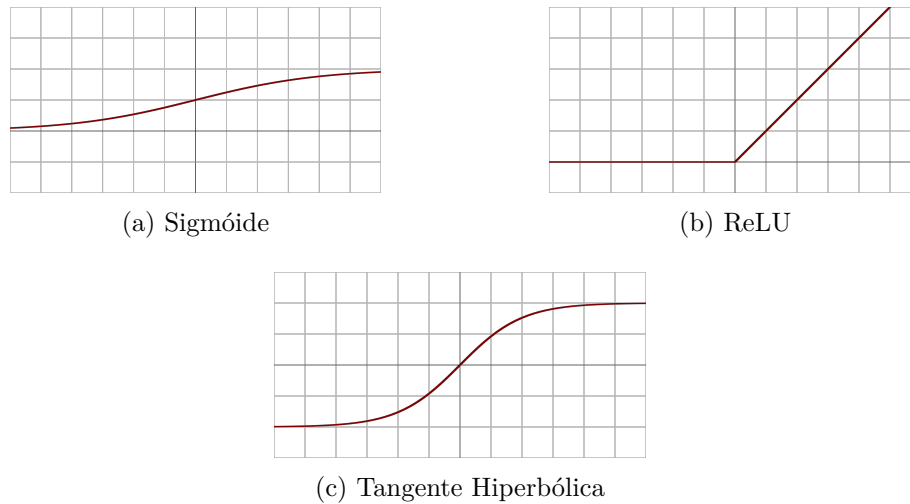


Figura 4.4: Funções de ativação

Por fim, o valor produzido pela função de ativação se torna a saída do neurônio. Como dito anteriormente, esse valor pode ser passado para outros neurônios, em sequência, ou até mesmo ser a saída definitiva da rede.

Para termos uma visão mais geral, podemos entender o processo de ativação da seguinte maneira:

1. O neurônio recebe várias entradas
2. Cada entrada é multiplicada por seu respectivo peso
3. Os produtos são somados junto ao *Bias*
4. A soma resultante passa por uma função de ativação para produzir a saída

O objetivo do treinamento de uma rede neural é ajustar os pesos e vieses para que eles se adequem ao conjunto de dados fornecidos gerando uma saída muito próxima da desejada. Esses ajustes podem ser feitos utilizando técnicas como o gradiente descendente e a propagação reversa (*backpropagation*) (WERBOS, 1990) (WATT; BORHANI; KATSAGGELOS, 2016).



### 4.3

#### Convolutional Neural Networks

AS CNNs, também conhecidas como Redes Neurais Convolucionais, são uma categoria de redes neurais que se mostrou particularmente eficaz para tarefas de processamento de imagem. As CNNs são inspiradas pela visão biológica e foram projetadas para reconhecer padrões diretamente dos pixels de imagens com mínimo pré-processamento. A operação central das CNNs é a convolução. Ao invés de conectar totalmente todos os neurônios da camada anterior, um neurônio está conectado apenas a um pequeno patch localizado na camada anterior, através de um objeto matemático chamado de filtro ou *kernel*. Um Kernel é uma matriz pequena, geralmente de tamanho  $n \times n$ , onde  $n$  é um número ímpar como 3, 5 ou 7. Este kernel é "deslizado" ao longo da imagem para produzir um novo mapa de características. (GOODFELLOW; BENGIO; COURVILLE, 2016) Em cada "deslizamento" ocorre uma operação de convolução, que pode ser definida da seguinte maneira:

Sejam  $f(\tau)$  e  $k(\tau)$  duas funções de domínio em  $\mathbb{R}$ , uma operação de convolução entre ambas pode ser escrita como:

$$f * k = \int_{-\infty}^{\infty} f(\tau)k(t - \tau) dt$$

Numa camada convolucional, podemos generalizar a expressão de "deslizamento" do *kernel*  $k$  por uma imagem  $I$  de tamanho  $m \times n$ , da seguinte forma:

$$k * I = Z(i, j) = \sum_m \sum_n k(m, n)I(i - m, j - n)$$

É importante ressaltar também que existem outras configurações que podemos alterar no *kernel* para torná-lo mais preciso de modificável. Uma das alterações que podemos fazer é em relação a quantidade de pixels que ele se move de cada vez, ou melhor, a quantidade de pixels que é desconsiderada em cada turno. Essa disposição é chamada de *stride* e está representada na imagem 4.5 (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outro conceito importante é o *padding*, representado na imagem 4.6, que consiste em adicionar zeros ao entorno da imagem, ou do mapa de característica para controlar a dimensão espacial do resultado, usado frequentemente para manter a mesma dimensão espacial após a convolução (GOODFELLOW; BENGIO; COURVILLE, 2016).

Um outro detalhe importante que precisa ser mencionado é sobre o canal de cores. Naturalmente, essas operações que foram descritas anteriormente são realizadas em apenas 1 dos canais da imagem, ou seja, se quisermos aplicar

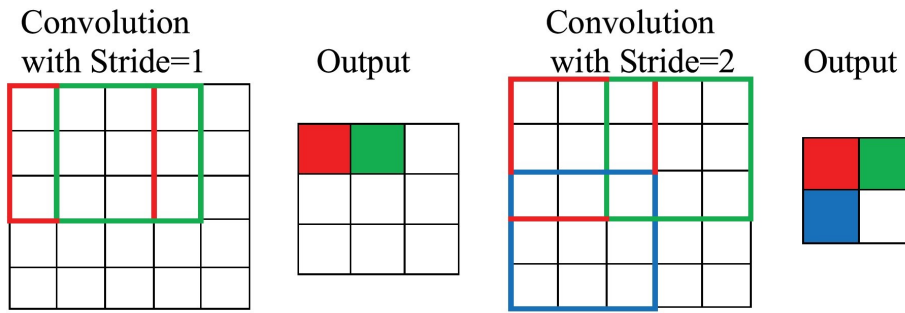


Figura 4.5: Exemplo de stride



Figura 4.6: Exemplo de padding

a metodologia apresentada, devemos repetir o processo para todos os canais **Red**, **Green** e **Blue** (WATT; BORHANI; KATSAGGELOS, 2016).

Após passar pela camada convolucional, sua saída é passada por uma função de ativação, como mostrado na seção de **Redes Neurais**. Nesse caso, normalmente é utilizada a função ReLU (*Rectified Linear Unit*) devido a sua eficácia (WATT; BORHANI; KATSAGGELOS, 2016).

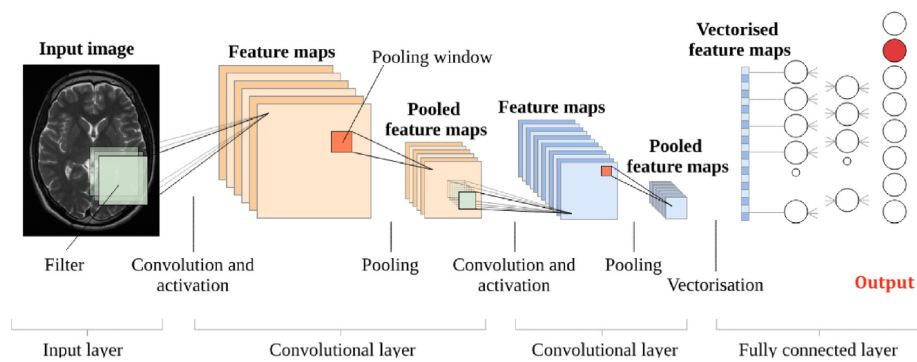


Figura 4.7: Visão completa de uma CNN

Na imagem 4.7 observamos todos os estágios para a criação de uma CNN, desde sua camada de *input* até sua saída. Percebemos também a presença de todas as camadas comentadas no texto, como as de convolução, *polling* e a camada totalmente conectada.

#### 4.4 ResNet50

A ResNet50 é uma variante da arquitetura de rede neural conhecida como Residual Network (ResNet), que foi uma das contribuições mais significativas no campo da visão computacional, particularmente no contexto de reconhecimento de imagens profundas. Desenvolvida por Kaiming He, Xiangyu Zhang, Shaoqing Ren e Jian Sun da Microsoft Research, estabelecendo um novo estado da arte para a classificação de imagens (HE et al., 2015).

O principal problema que as conexões residuais visam resolver é o do "desaparecimento do gradiente", que é um desafio comum ao treinar redes neurais profundas. À medida que a rede se aprofunda, os gradientes (que são essenciais para a atualização dos pesos durante o treinamento) podem se tornar tão pequenos que não têm efeito perceptível, tornando a rede difícil ou impossível de treinar. As conexões residuais permitem que o sinal de gradiente seja diretamente propagado para camadas anteriores sem passar por todas as transformações de camadas intermediárias. Isso é feito adicionando a saída de uma camada anterior à saída de uma camada mais avançada, efetivamente permitindo que a rede aprenda a função de identidade se necessário. Isso significa que, em vez de aprender uma transformação direta, as camadas estão aprendendo um resíduo (HE et al., 2015).

A ResNet50 é estruturada em quatro estágios principais, contendo uma série de blocos residuais, com o número de blocos em cada estágio sendo 3, 4, 6 e 3, respectivamente. À medida que a rede progride através dos estágios, o tamanho das características é reduzido e o número de filtros é aumentado para manter a carga computacional gerenciável (SHARMA et al., 2022).

Após todos os blocos residuais, a rede aplica uma camada de average pooling para condensar as características espaciais em um único vetor por característica. Isso é seguido por uma camada totalmente conectada que mapeia as características para o número desejado de classes e uma camada softmax para calcular as probabilidades de classificação (SHARMA et al., 2022).

Durante o treinamento, as conexões residuais facilitam a otimização da rede, mesmo com sua profundidade significativa. A ResNet50 pode ser treinada a partir do zero ou usada como uma rede pré-treinada em grandes conjuntos

de dados, como o ImageNet, e ajustada para tarefas específicas através da transferência de aprendizado.

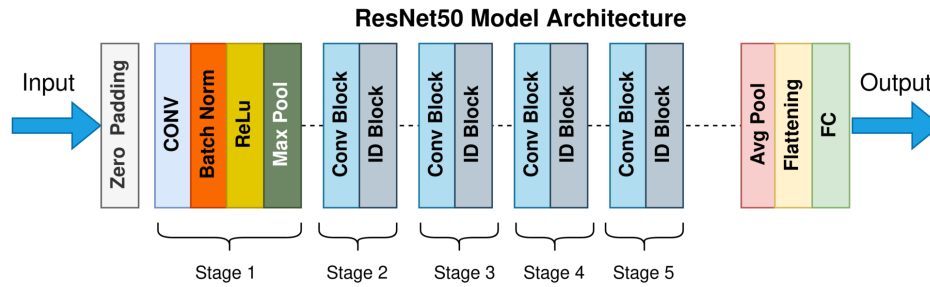


Figura 4.8: Arquitetura da ResNet50

Como representado na arquitetura da ResNet na imagem 4.8, a rede começa com uma única camada de convolução seguida por uma camada de max pooling, depois segue para uma série de blocos residuais e termina com uma camada de average pooling, uma camada totalmente conectada e uma função de ativação, como por exemplo a softmax, para tarefas de classificação.

A ResNet50, devido à sua profundidade e eficiência, tornou-se uma escolha popular para muitas aplicações de visão computacional, incluindo reconhecimento de objetos, detecção de objetos e segmentação semântica. Ela também é frequentemente usada como uma rede pré-treinada para transferência de aprendizado, onde uma rede treinada em um grande conjunto de dados (como o ImageNet) é adaptada para uma tarefa específica com um conjunto de dados menor (SHARMA et al., 2022).

O sucesso da ResNet e suas variantes inspirou muitas outras arquiteturas de redes neurais e continua a ser uma área ativa de pesquisa e aplicação. A capacidade de treinar redes profundas sem o problema do desaparecimento do gradiente abriu novas possibilidades para o design de arquiteturas de rede neural e ajudou a avançar o campo da inteligência artificial. Podemos entender mais até sobre o modelo e sua forma como foi construído no repositório oficial da biblioteca TensorFlow, utilizada nessa tese, acesse **Tensorflow: ResNet50**

## 4.5 InceptionV3

Outro modelo de Rede Neural mundialmente utilizado é a InceptionV3. Ele foi introduzido pela primeira vez em 2014 por pesquisadores da Google na sua primeira versão (InceptionV1 ou GoogLeNet), após ter ganhado o desafio de classificação de imagem *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* (SZEGEDY et al., 2015b).

O modelo Inception v1 introduziu o conceito de módulos Inception. Eles foram criados com o objetivo de tornar as redes neurais convolucionais mais adaptáveis e capazes de extrair informações de uma vasta gama de entradas de imagem. A funcionalidade desses módulos se baseia em realizar várias operações de convolução simultaneamente, cada uma com filtros de tamanhos distintos, como 1x1, 3x3 e 5x5. Essa variedade permite que a rede detecte desde os detalhes mais sutis até as características mais amplas e abstratas das imagens. Em paralelo às convoluções, os módulos também executam operações de pooling, geralmente do tipo max pooling, que contribuem para a redução da dimensionalidade dos dados e ajudam a rede a se tornar invariante a pequenas mudanças e deslocamentos na imagem. Após realizar as convoluções e o pooling, a rede combina as saídas de todas essas operações paralelas, concatenando-as. Essa concatenação é feita ao longo do eixo do canal, garantindo que a riqueza de informações capturadas em cada processo seja mantida e transmitida adiante na rede (SZEGEDY et al., 2015a).

Para evitar um crescimento exponencial no número de cálculos necessários, devido ao aumento das dimensões, os módulos Inception aplicam uma técnica de redução de dimensão. Isso é feito através do uso de convoluções 1x1 antes de aplicar filtros maiores, o que diminui a profundidade do volume de entrada sem perder informações essenciais. Com essa abordagem no modelo, podemos garantir que a rede neural mantenha sua eficiência computacional, mesmo enquanto processa grandes quantidades de dados (SZEGEDY et al., 2015a).

Após as inovações da sua primeira versão, foi lançada uma V2 que trouxe uma melhoria na sua arquitetura original, incluindo otimizações para reduzir mais ainda o consumo operacional. Uma das melhorias inseridas foi a adição da *batch normalization*, que por consequência, acelerou o treinamento do modelo (INCEPTION... , 2023).

A Inception v3 trouxe avanços notáveis em relação às suas versões antecessoras, mantendo as otimizações da Inception v2 e adicionando novas melhorias. A arquitetura, como representada na imagem 4.9, refinou os módulos Inception com ajustes que padronizam os tamanhos dos filtros de convolução, resultando em uma redução substancial do número de operações computacionais necessárias. Além disso, a Inception v3 substituiu as convoluções de maior dimensão por uma sequência de duas convoluções de 3x3, diminuindo assim o número de parâmetros e o custo computacional envolvido (SZEGEDY et al., 2016).

Como citado anteriormente, a arquitetura também adotou o uso de convoluções assimétricas, empregando uma combinação de convoluções 1xN segui-

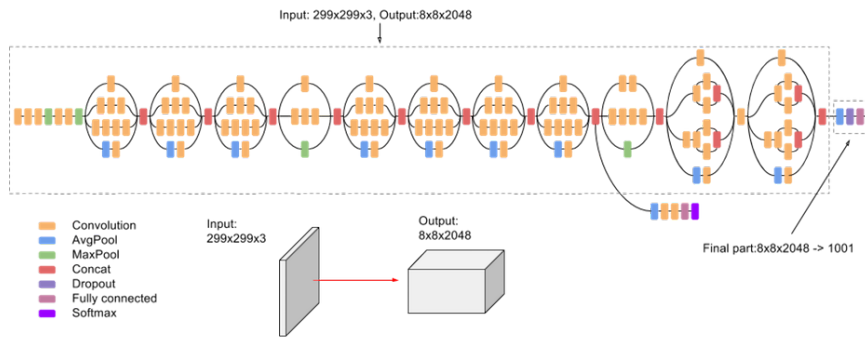


Figura 4.9: Arquitetura da InceptionV3

das por  $N \times 1$ , uma estratégia eficaz para substituir as convoluções quadradas tradicionais e reduzir ainda mais as demandas computacionais sem comprometer a eficiência na detecção de características. Para fortalecer a capacidade do modelo de generalizar a partir de conjuntos de dados de treinamento, a Inception v3 integrou técnicas de ampliação de dados mais avançadas, aumentando a robustez do modelo ao expô-lo a uma variedade mais ampla de variações de imagem durante o treinamento (SZEGEDY et al., 2015a), (SZEGEDY et al., 2016). Podemos entender mais até sobre o modelo e sua forma como foi construído no repositório oficial da biblioteca TensorFlow, utilizada nessa tese, acesse **Tensorflow: Inception**

## 4.6

### Otimizador RMSprop

Algoritmos de otimização como desempenham um papel crucial no treinamento de redes neurais e outros modelos de aprendizado de máquina, pois aceleram a convergência para o mínimo da função de custo, o que é determinante para a avaliação do desempenho do modelo. Essa aceleração resulta em economia de tempo e recursos computacionais. Além disso, esses algoritmos previnem que o modelo fique retido em mínimos locais ou pontos de sela, garantindo assim que as soluções encontradas sejam robustas e representativas das melhores configurações possíveis (WATT; BORHANI; KATSAGGELOS, 2016).

A capacidade de ajustar hiperparâmetros como a taxa de aprendizado é outra vantagem significativa, já que alguns algoritmos adaptam esses parâmetros dinamicamente durante o treinamento, potencialmente levando a resultados superiores. Eles são particularmente eficientes em grandes conjuntos de dados, onde calcular o gradiente para todos os exemplos é impraticável, optando por atualizar os parâmetros usando mini-lotes de dados (WATT; BORHANI;

KATSAGGELOS, 2016).

A adaptabilidade desses algoritmos também permite que se ajustem a uma variedade de problemas, especialmente aqueles com superfícies de erro irregulares, suavizando as oscilações e promovendo atualizações mais consistentes. Eles mantêm a estabilidade numérica durante o treinamento, evitando extremos que poderiam resultar em problemas computacionais. Finalmente, ao otimizar a função de custo, contribuem para a capacidade de generalização do modelo, melhorando seu desempenho em dados novos e, portanto, sua utilidade em aplicações do mundo real (WATT; BORHANI; KATSAGGELOS, 2016).

Um dos algoritmos de otimização utilizado é o RMSProp, abreviação de "*Root Mean Square Propagation*". Ele é um algoritmo de otimização projetado especificamente para treinar redes neurais artificiais (ANNs) e foi proposto por Geoff Hinton durante um curso online sobre Redes Neurais para Aprendizado de Máquina (ELSHAMY et al., 2023).

Este algoritmo é uma adaptação do *Stochastic Gradient Descent* (SGD) e faz parte dos métodos de taxa de aprendizagem adaptativa. O principal objetivo do RMSProp é ajustar de forma mais eficaz a taxa de aprendizagem durante o processo de treinamento. Para fazer isso, mantém uma média móvel (descontada) do quadrado dos gradientes e usa essa média para normalizar o gradiente ao atualizar os pesos da rede. Ou seja, ajusta a taxa de aprendizagem para cada parâmetro individualmente, em vez de ter uma taxa de aprendizagem fixa para todos os parâmetros (ELSHAMY et al., 2023). Podemos escrever os cálculos de atualização da seguinte forma:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1 - \beta) \left( \frac{\partial C}{\partial w} \right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\partial C}{\partial w}$$

Onde,  $E[g]$  é a média móvel dos quadrados dos gradientes,  $\frac{\partial C}{\partial w}$  é o gradiente da função de custo com o respectivo peso ( $w$ ); A constante  $\eta$  é a taxa de aprendizado e  $\beta$  é o parâmetro da média móvel, sendo normalmente utilizado como 0.9 no seu valor.

## 4.7

### Otimizador Adam

O otimizador Adam, abreviação de "*Adaptive Moment Estimation*", é um algoritmo para otimização que é usado amplamente no treinamento de redes neurais e outras aplicações de aprendizado de máquina. Ele foi introduzido

por Kingma e Ba em 2014 e é particularmente eficiente em casos com grandes volumes de dados e parâmetros, como mostram problemas envolvendo imagem (WATT; BORHANI; KATSAGGELOS, 2016).

Adam combina ideias do "*momentum*" e do "RMSprop". No "*momentum*", utiliza-se a média exponencialmente ponderada dos gradientes para acelerar o gradiente descendente. Esta técnica ajuda o algoritmo a acelerar em direções consistentes, o que auxilia na convergência mais rápida para mínimos. O RMSprop, por outro lado, adapta a taxa de aprendizado para cada parâmetro ao calcular uma média móvel exponencial dos quadrados dos gradientes. Isso permite ajustes mais finos do aprendizado, evitando passos demasiadamente grandes ou pequenos que poderiam prejudicar a convergência (GESS; KASSING, 2023).

O otimizador Adam, portanto, herda as vantagens de ambos os métodos, oferecendo uma abordagem que consegue lidar bem com a variação nos gradientes e a escala dos dados. Isso é feito mantendo estimativas separadas dos primeiros momentos (o equivalente ao momentum) e dos segundos momentos (o equivalente ao RMSprop) dos gradientes (GESS; KASSING, 2023).

O aspecto matemático do Adam envolve o cálculo das médias móveis exponencialmente ponderadas dos gradientes e dos quadrados dos gradientes. Entretanto, como essas médias são inicializadas como zero, o otimizador faz uma "correção de viés" para ajustar e escalar as médias móveis, garantindo que elas sejam imparciais. Intuitivamente, o Adam ajusta a descida do gradiente após cada iteração para mantê-la sob controle e imparcial durante todo o processo de otimização (WATT; BORHANI; KATSAGGELOS, 2016).

Em termos de desempenho, o Adam é conhecido por superar muitos outros otimizadores, proporcionando um treinamento eficiente e eficaz, com menor custo computacional e melhor desempenho em muitos cenários. O ajuste fino que o Adam proporciona, ao controlar a taxa de aprendizado de maneira adaptativa, permite que ele navegue eficientemente através de paisagens complexas de otimização, alcançando bons mínimos em tempo hábil e com estabilidade (GESS; KASSING, 2023).

A imagem 4.10, mostra um comparativo entre diversos otimizadores aplicados à tarefa de classificar imagens do conjunto de dados **MNIST**, torna-se evidente que, apesar de todos visarem a minimização do erro, o Adam sobressai significativamente em termos de desempenho ao longo das iterações. Em contrapartida, o RMSProp, mencionado previamente, exibe um desempenho consistente, mas não extraordinário, situando-se em um patamar intermediário em relação aos demais otimizadores analisados.



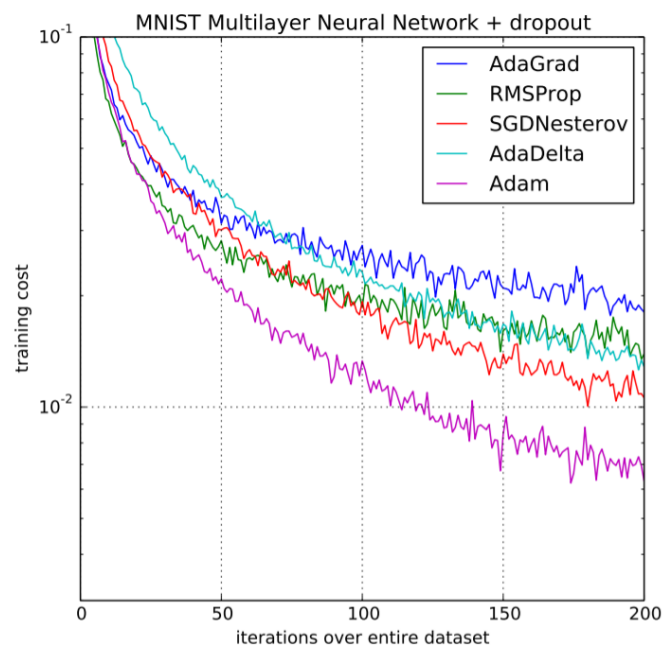


Figura 4.10: Performance entre otimizadores

## 5

# Projeto e Especificação do Sistema

Nesse capítulo serão abordadas mais informações sobre as bibliotecas necessárias para criação e treinamento dos modelos, detalhando cada uma de suas funções e métodos utilizados. Será apresentado também uma descrição detalhada do dataset utilizado para o treinamento como também seus metadados.

### 5.1

#### Bibliotecas

Para a confecção do projeto, foram utilizadas uma série de bibliotecas para a criação, teste, otimização e avaliação dos modelos de rede neural.

---

#### Código 1: Bibliotecas utilizadas

---

```
1 import pandas as pd
2 import numpy as np
3
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 import keras
7 from keras.models import Sequential
8 from keras.layers import (
9     Dense,
10    Conv2D,
11    MaxPool2D,
12    Flatten,
13    Dropout,
14    BatchNormalization
15 )
16 from keras.preprocessing.image import ImageDataGenerator
17 from sklearn.model_selection import train_test_split
18 from sklearn.metrics import (
19     classification_report,
20     confusion_matrix,
21     ConfusionMatrixDisplay
22 )
23 )
24 from keras.callbacks import ReduceLROnPlateau
25 import tensorflow as tf
26 import tensorflow.keras.backend as K
27
28 import cv2
29 import os
```

---

Primeiramente, **Matplotlib** e **Seaborn** são usadas para criar gráficos e visualizações. **Matplotlib** é a escolha tradicional para gráficos estáticos e interativos, enquanto Seaborn oferece uma abordagem mais estilizada para visualizações de dados.

O **Keras**, operando em cima da biblioteca **TensorFlow**, é uma API de alto nível para construir e treinar modelos de aprendizado profundo. Com essa biblioteca, é possível empilhar camadas de redes neurais usando o modelo **Sequential** e adicionar uma variedade de camadas como **Dense** para conexões densas, **Conv2D** para convolução em dados de imagem, **MaxPool2D** para pooling, **Flatten** para achatar as dimensões dos dados, **Dropout** para reduzir o overfitting e **BatchNormalization** para normalizar as ativações dos neurônios. O **ImageDataGenerator** é uma ferramenta poderosa para aumentar o conjunto de dados de imagem e melhorar o desempenho do modelo. Além disso, o **ReduceLROnPlateau** é um callback que ajusta a taxa de aprendizado durante o treinamento para otimizar o processo.

O pacote **Scikit-learn** foi empregado para dividir o conjunto de dados em treino e teste com **train\_test\_split** e avaliar o desempenho do modelo com métricas como **classification\_report** e **confusion\_matrix**, além de fornecer uma visualização clara do desempenho com **ConfusionMatrixDisplay**. Vale ressaltar que a versão utilizada do pacote (2.1) ainda não possuía o método **ConfusionMatrixDisplay**, foi usado o **pandas** aliado com o **seaborn** para fazer tal visualização.

É importante dizer que o **TensorFlow** é a base sobre a qual o **Keras** é construído, e o *backend* do **Keras** permite operações de baixo nível que são independentes da plataforma, o que significa que é possível escrever código que é compatível com diferentes frameworks de aprendizado de máquina.

O **OpenCV** é uma biblioteca robusta para processamento de imagens e visão computacional, que permite manipular e processar imagens para tarefas como reconhecimento facial ou detecção de objetos, nesse caso, aplicado para leitura e carregamento dos conjuntos de dados. O módulo **os** foi utilizado para interagir com o sistema de arquivos do sistema operacional, permitindo operações como a leitura e escrita de arquivos.

Juntas, essas bibliotecas e funções formam um ecossistema completo para o desenvolvimento de projetos de aprendizado de máquina, desde a preparação e visualização de dados até a construção, treinamento e avaliação de modelos complexos, nesse caso, relacionados ao processamento de imagens médicas.

## 5.2

### Análise do Dataset

Esta tese apresenta redes neurais projetadas para determinar a presença de pneumonia a partir de radiografias torácicas. Essas imagens de raio-X (anterior-posterior) foram selecionadas de retrospectivas de pacientes pediátricos de um a cinco anos de idade do Centro Médico de Mulheres e Crianças de Guangzhou, China, sendo todas elas parte de um atendimento clínico de rotina.

Para a análise das imagens de raio-X, todas as radiografias foram inicialmente examinadas para controle de qualidade, removendo todas as varreduras de baixa qualidade. Os diagnósticos para as imagens foram então classificados por dois médicos especialistas antes de serem liberados para treinar o sistema de IA. A fim de levar em conta quaisquer erros de classificação, o conjunto de avaliação também foi verificado por um terceiro especialista.

O conjunto de dados usado para treinar e testar essas redes pode ser obtido no Kaggle sob o nome "**Chest X-Ray Images (Pneumonia)**", ou no próprio artigo onde utilizaram esse dataset (KERMANY; ZHANG; GOLDBAUM, 2018). No total existem 5866 imagens, que estão divididas em três diretórios distintos:

1. **test**: Representando o conjunto de dados para testagem do modelo após ter sido treinado, contendo 234 imagens de pulmões saudáveis e 390 com pneumonia.
2. **train**: Representando o conjunto de imagens que serão utilizadas para o treinamento do modelo, contendo 1351 imagens de pulmões saudáveis e 3875 imagens de pulmões que contrairam a doença.
3. **val**: Representando as imagens que serão utilizadas durante o treinamento do modelo, não para ajustar os pesos do modelo, e sim para servir como uma espécie de validação das previsões, monitorando a aprendizagem e generalização do modelo, ajudando a indentificar problemas como overfitting. Nesse caso um volume menor de imagens, com 8 de cada categoria.

É importante ressaltar também que as imagens encontradas nessa base estão no formato `.jpeg`, na escala de cores cinza e elas não possuem um tamanho fixo.

## 6

## Implementação e Avaliação

Nesse capítulo serão abordados todas as etapas da processamento, criação/treinamento e avaliação dos 3 modelos selecionados. Na etapa de treinamento será explicado um pouco do processamento das imagens, na fase criação/treinamento será exemplificado melhor o padrão de construção dos modelos e, por fim, na fase de avaliação serão exibidos os resultados encontrados.

### 6.1

#### Análise e Processamento dos Dados

Como apontado no capítulo 5, o dataset consiste numa série de imagens de raio-X de pulmões. As imagens têm extensão `.jpeg`, não possuem um tamanho fixo e estão separadas em 3 pastas (`train`, `test` e `val`) de acordo com suas devidas funções. Por isso, antes de começar o treinamento e avaliação dos modelos de rede neural, foi necessário passar por uma etapa de processamento, onde realizaremos transformações no conjunto de dados, seguindo o algoritmo 1.

---

**Algoritmo 1:** Processamento dos dados

---

**Entrada:** Caminho de dados

**Saída:** Vetor com dados carregados

```
1 resultado ← cria_lista_vazia()
2 para todo classe faça
3   caminho ← monta_caminho(Caminho de dados, classe)
4   acessa_caminho(caminho)
5   para todo imagem faça
6     carrega_imagem()
7     redimensiona_imagem()
8   atualiza_vetor(resultado, [imagem, classe])
9 Retorna resultado
```

---

O algoritmo 1 recebe de entrada um diretório onde ele deverá ler as imagens e retorna um vetor com as imagens carregadas. Esse algoritmo será executado para todos os 3 diretórios (`train`, `test` e `val`), onde ele percorrerá por todas as imagens redimensionando para o tamanho ideal (nesse caso sendo 150 pixels por 150 pixels). Por fim ele atualiza o vetor de saída com um novo vetor contendo a imagem transformada e sua classe (NORMAL ou PNEUMONIA).

Utilizando a função criada, podemos carregar as imagens para 3 cenários diferentes (`train`, `test` e `val`) e analisar o conjunto. Um exemplo de imagens lidas aplicando a paleta de cores `bone` está representada na imagem 6.1.

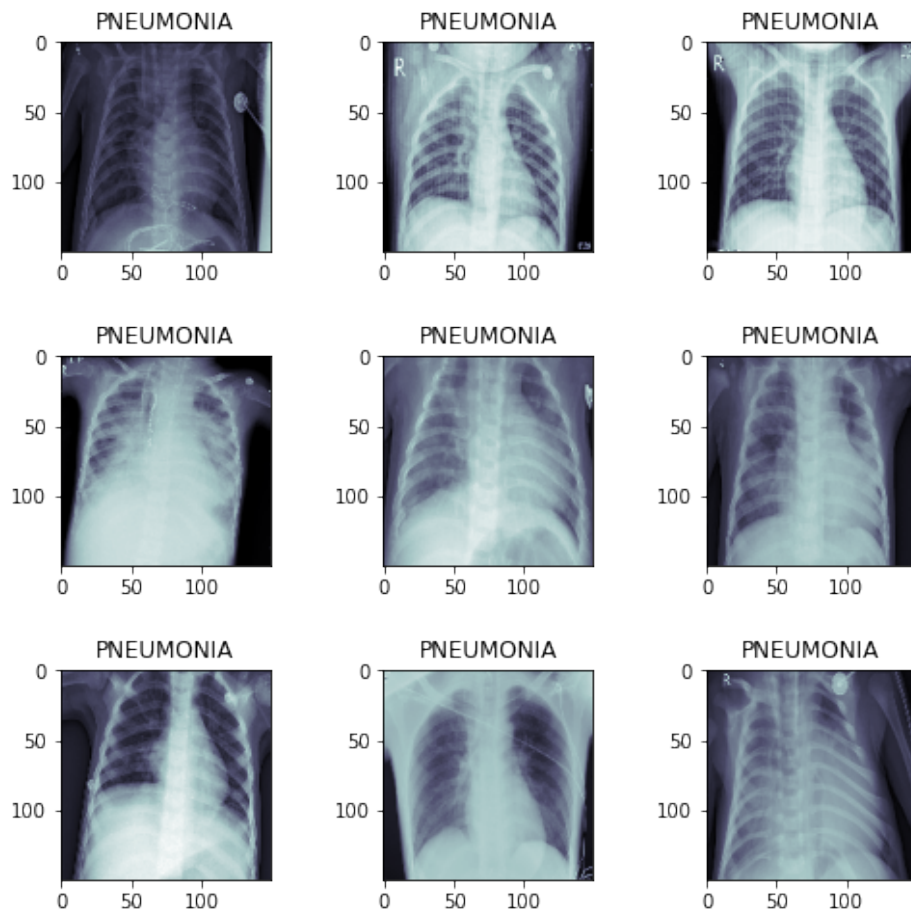


Figura 6.1: Raio-X de pulmões

Colocando as imagens em um dataframe da biblioteca `pandas`, conseguimos ter uma visualização melhor e exibir gráficos envolvendo a quantidade de cada classe em cada estágio do processo.

Na imagem 6.2, é trivial perceber que as classes estão desbalanceadas no conjunto de dados, onde a barra vermelha representa as imagens de pulmões com pneumonia e a barra azul de pulmões saudáveis, tendo seus estágios de treino, teste e validação representando 89%, 10% e 0.2% da base respectivamente. Sendo assim, será necessário aplicar uma etapa de enriquecimento de dados (*data augmentation*) para gerar novas amostras para cada estágio.

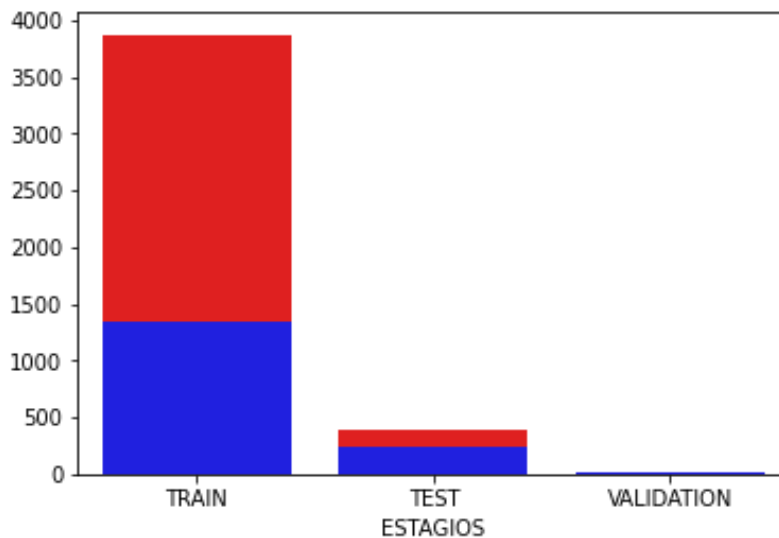


Figura 6.2: Dados dos estágios

Para seguir com o passo do enriquecimento de dados, utilizamos uma função da biblioteca `keras` chamada de `ImageDataGenerator` que aceita uma série de parâmetros para gerar diferentes variações nas imagens. Para o treinamento e validação do modelo foram utilizados diferentes configurações:

1. As imagens foram rotacionadas em  $30^\circ$ , sendo útil para reconhecer objetos que podem aparecer em várias orientações.
2. As imagens foram transladadas ligeiramente para cima e para lado em 10% do seu valor, com o objetivo de reconhecer traços de pneumonia mesmo não estando centralizado na imagem
3. Um zoom de 20% foi aplicado tendo como finalidade reconhecer objetos de diferentes escalas.
4. Um *flip* horizontal, já que a orientação horizontal do raio-X não importa. Nesse caso, não tem problema trocar pulmões esquerdos por direitos, pois o objetivo do modelo é puramente binário.

Outra alteração necessária a ser feita foi a normalização dos valores dos pixels. Nas imagens atuais, os valores de cores estão entre 1 e 255, porém um modelo de CNN performa melhor se essa grandeza estiver entre 0 e 1.

## 6.2

### Criação dos Modelos

Uma função específica foi desenvolvida para construir e treinar modelos seguindo um formato padrão. Utiliza-se a função *Sequential* da biblioteca *keras* para definir o modelo, sobre a qual são empilhadas diversas camadas necessárias. Estas podem incluir convoluções, operações de *pooling*, normalizações, entre outras. Todas as funções empregadas na criação dos modelos aderem a este padrão estabelecido exemplificado no algoritmo 2

---

#### Algoritmo 2: Criação e treinamento de um modelo

---

**Entrada:** Número de épocas, tamanho do lote, dados de treino e dados de validação

**Saída:** Histórico de treinamento e modelo treinado

```

1 modelo ← Sequential
2 modelo.empilha_camadas_necessárias()
3 modelo.compile(otimizador=otimizador_escolhido)
4 histórico ← modelo.treinar(dados de treino, dados de validação, número
   de épocas, tamanho do lote)
5 Retorna histórico, modelo
```

---

Como especificado na 4, os algoritmos de rede neural ResNet50 e InceptionV3, foram treinados com imagens com 3 canais de cores. No caso do *dataset* utilizado, não será necessário aplicar técnica alguma para "colorir" as imagens de escala cinza, basta apenas repetir a mesma camada cinza 3 vezes ao enviar as imagens para o modelo.

## 6.3

### Resultados

Depois de treinar os modelos, avaliamos sua eficácia usando várias métricas, incluindo acurácia, perda durante as fases de treinamento e teste, matrizes de confusão e relatórios de classificação. Experimentos variados foram realizados, empregando otimizadores como **Adam** e **RMSPprop** em combinação com modelos como InceptionV3, ResNet50, e uma rede neural convolucional. Os treinamentos foram realizados com 20 épocas, gerando os resultados encontrados na tabela 6.1.



Modelo	Otimizador	Precisão	Perda	Acurácia	f1-score	Recall	Miss
CNN	RMSProp	0.88	0.54	87.82%	0.88	0.88	76
ResNet50	RMSProp	0.91	0.23	90.86%	0.91	0.91	57
InceptionV3	RMSProp	0.94	0.15	94.23 %	0.94	0.94	36
CNN	Adam	0.89	0.27	88.94%	0.89	0.89	69
ResNet50	Adam	0.93	0.17	93.42%	0.93	0.93	41
InceptionV3	Adam	0.95	0.16	95.19%	0.95	0.95	30

Tabela 6.1: Resultados do treinamento

A precisão é determinada pela proporção de predições acertadas para uma determinada classe em relação ao total de predições feitas para essa classe. Já o recall mede a proporção de predições acertadas para uma classe em comparação com o número total de casos verdadeiros dessa classe no conjunto de dados. O F1-Score, por sua vez, é calculado como a média harmônica da precisão e do recall, resultando em uma métrica única que balanceia as duas. Depois de calcular essas métricas para cada classe, realiza-se uma média delas para chegar aos valores apresentados na tabela.

Analisando mais detalhadamente a tabela 6.1 é perceptível que, quando foi feita a troca do otimizador RMSProp para Adam tivemos um aumento de até 2 pontos percentuais na acurácia, recall, f1-score e precisão, assim como a queda nesse caso, bem mais significativa, da perda. Ambos cenários indicam resultado positivos para esse dataset.

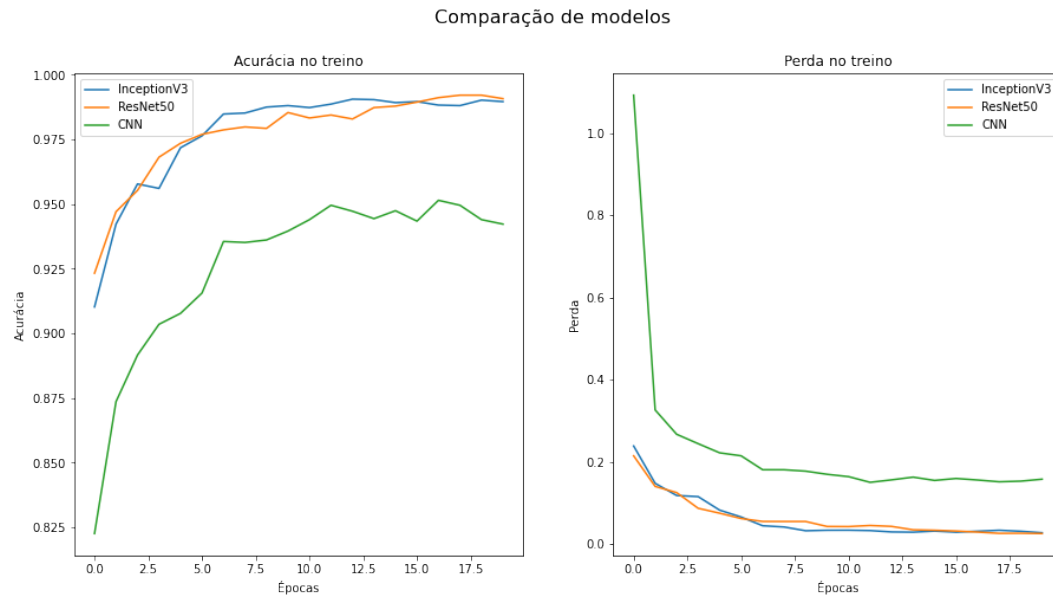


Figura 6.3: Comparação de acurácia e perda

Analisando a imagem 6.3, percebemos que a curva da CNN se distancia das outras, mantendo uma acurácia menor e uma perda maior. Isso se dá devido ao nível de simplicidade e à ausência de pesos pré-treinados, como é o caso da InceptionV3 e da ResNet50. Falando um pouco sobre esses dois últimos modelos, é notável que ambos ficam muito próximos tanto no quesito de minimizar a perda quanto a maximizar a acurácia. De um modo geral, a rede neural InceptionV3 se sai melhor do que a ResNet50, pelo fato de convergir para seus valores ideais de maneira mais "rápida".

Analisando as matrizes de confusão para cada um dos modelos, são notados alguns resultados interessantes:

		Valor previsto		Total
		Pneumonia	Normal	
Valor real	Pneumonia	356	34	390
	Normal	35	199	234
Total		391	233	624

Tabela 6.2: CNN - Matriz de confusão

Para os resultados previstos incorretamente, o pior caso, seriam os falsos positivos. Na CNN genérica montada eles representam aproximadamente 5.4% do total de previsões, como mostra a tabela 6.2.

		Valor previsto		Total
		Pneumonia	Normal	
Valor real	Pneumonia	373	17	390
	Normal	24	210	234
Total		397	227	624

Tabela 6.3: ResNet50 - Matriz de confusão

Implementando um modelo mais robusto, é notória a diferença, representada na tabela 6.3, em ambos os casos incorretos com falsos positivos caindo 50% e falsos negativos caindo 25%.

		Valor previsto		Total
		Pneumonia	Normal	
Valor real	Pneumonia	382	8	390
	Normal	22	212	234
Total		404	220	624

Tabela 6.4: InceptionV3 - Matriz de confusão

Por fim, no InceptionV3 são encontrados os menores valores de classificações incorretas em todo o teste. Podemos ver na tabela 6.4 que, o total de falsos positivos decresceu 76% em relação ao valor original, e os falsos negativos, um total de 37%.

Com base no que foi apresentado, fica evidente que o modelo InceptionV3 supera os demais em eficácia, otimizando a precisão e reduzindo significativamente as perdas. Esse modelo se destaca especialmente na matriz de confusão ao registrar o menor número de falsos positivos e negativos. Essa característica é crucial, particularmente em análises médicas de doenças, onde um resultado falso negativo pode ter consequências graves para a saúde do paciente. Portanto, a eficiência do InceptionV3 em minimizar esses erros contribui diretamente para a confiabilidade e segurança dos diagnósticos.

## 7

### Considerações Finais

Como lido anteriormente, a questão do uso de inteligência artificial aplicada na área de imagens médicas é um ganho positivo para profissionais de saúde. Existindo um modelo pré-treinado é possível:

1. Melhorar a precisão do diagnóstico, podendo analisar imagens com padrões de doença sutis demais para detecção humana, como citada na introdução o caso da detecção do cancer de mama 5 anos antes de se formar completamente.
2. Reduzir tempo de diagnósticos, analisando grandes volumes de dados de imagem rapidamente, tornado-se crucial para situações de emergência.
3. Minimizar erros humanos que podem ocorrer devido á fadiga viés ou simplesmente limitações humanas na interpretação de imagens complexas.

Com os modelos testados nessa tese, foi possível chegar à um resultado bem positivo de acurácia, mesmo tendo uma base de dados completamente enviesada para casos de pneumonia. O modelo que se mais destacou foi o **InceptionV3** aliado com o otimizador **Adam**, apresentando uma acurácia de 95.19%, com pequenas incidências de erros.

Para os próximos passos do projeto, seria possível treinar os modelos (CNN, ResNet50 e InceptionV3) em uma base balanceada entre as classes de pneumonia e saudável, gerando uma confiabilidade maior no modelo e por consequência diminuindo as ocorrências de falsos negativos nos resultados. Seria possível também, aplicar a mesma lógica de classificação à outros de tipos de casos como doenças oculares, cardíacas, ósseas/musculares e até mesmo para detecção de tumores.

Embora os benefícios sejam consideráveis, substituir médicos e profissionais da saúde exclusivamente por inteligências artificiais não é viável. Isso ocorre porque, apesar da menor incidência de erros, esses modelos ainda são suscetíveis a equívocos ocasionais. Como ilustrado pelas matrizes de confusão no capítulo anterior, mesmo reduzindo ao máximo os casos de falsos negativos, houve 22 pacientes que correram o risco de receber um diagnóstico falso de não terem pneumonia, quando na verdade a possuíam. Erros desse tipo podem ameaçar a vida dos pacientes e de outros.

Dessa forma, é fundamental reconhecer o papel insubstituível da experiência humana e do julgamento clínico na área da saúde. A inteligência artificial

deve ser vista como uma ferramenta poderosa que pode melhorar a eficiência e a precisão dos diagnósticos, mas sempre complementando e trabalhando em conjunto com as habilidades e conhecimentos dos profissionais de saúde. O equilíbrio entre tecnologia e atendimento humano é essencial para garantir uma assistência médica de alta qualidade, centrada no paciente e adaptada às complexidades e variabilidades inerentes ao campo da saúde.

## Referências bibliográficas

ANTHIMOPOULOS, M. et al. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1207–1216, 2016.

AREVALO, J. et al. Representation learning for mammography mass lesion classification with convolutional neural networks. **Computer Methods and Programs in Biomedicine**, v. 127, p. 248–257, 2016. ISSN 0169-2607. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169260715300110>>.

BRASIL, M. da Saúde do. **Dia Mundial da Pneumonia**. 2023. <<https://bvsms.saude.gov.br/12-11-dia-mundial-da-pneumonia/>>. Acessado em: 2023-11-17.

BROSCH, T. et al. Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1229–1239, 2016.

CHUN, M. **How Artificial Intelligence is Revolutionizing Drug Discovery**. Bill and Health, 2023. Disponível em: <<https://blog.petrieflom.law.harvard.edu/2023/03/20/how-artificial-intelligence-is-revolutionizing-drug-discovery/#:~:text=De%20novo%20drug%20design%3A%20While,drug%20molecules%20entirely%20from%20scratch.>>

CONNER-SIMONS, R. G. A. Using ai to predict breast cancer and personalize care. Massachusetts Institute of Technology, 2019. Disponível em: <<https://news.mit.edu/2019/using-ai-predict-breast-cancer-and-personalize-care-0507>>.

CONTROL, C. for D.; PREVENTION. **Symptoms and Complications - Pneumococcal Disease**. 2022. <<https://www.cdc.gov/pneumococcal/about/symptoms-complications.html>>. Acessado em: 2023-11-17.

DADONAITE, B.; ROSER, M. **Pneumonia**. 2019. <<https://ourworldindata.org/pneumonia>>. Acessado em: 2023-11-17.

DASGUPTA, A.; SINGH, S. A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation. In: **2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)**. [S.l.: s.n.], 2017. p. 248–251.

ELSHAMY, R. et al. Improving the efficiency of rmsprop optimizer by utilizing nestrovo in deep learning. **Scientific Reports**, Nature Publishing Group, v. 13, n. 1, 2023. Disponível em: <<https://www.nature.com/articles/s41598-023-35663-x>>.

FANDI, G. et al. The impact of cyber attack on emergency energy system. **American Journal of Electrical Power and Energy Systems**, v. 11, n. 6, p. 118–122, 2023. Disponível em: <<https://doi.org/10.11648/j.epes.20221106.12>>.

FILHO, S. Como a inteligência artificial vai cuidar da sua saúde. 2018. Disponível em: <<https://www.istoedinheiro.com.br/como-a-inteligencia-artificial-vai-cuidar-da-sua-saude/>>.

FROES, N. R. F.; BLASI, F. Pneumococcal vaccination and chronic respiratory diseases. **International Journal of Chronic Obstructive Pulmonary Disease**, Dove Medical Press, v. 12, p. 3457–3468, 2017. PMID: 29255353. Disponível em: <<https://www.tandfonline.com/doi/abs/10.2147/COPD.S140378>>.

GAO, M. et al. Image super-resolution based on two-level residual learning cnn. **Multimedia Tools and Applications**, v. 79, n. 7–8, p. 4831–4846, 2018.

GAO ULAS BAGCI, L. L. A. W. M. B. H.-C. S. H. R. G. Z. P. A. D. R. M. S. Z. X. M.; MOLLURA, D. J. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. **Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization**, Taylor Francis, v. 6, n. 1, p. 1–6, 2018. PMID: 29623248. Disponível em: <<https://doi.org/10.1080/21681163.2015.1124249>>.

GESS, B.; KASSING, S. Convergence rates for momentum stochastic gradient descent with noise of machine learning type. **arXiv preprint arXiv:2302.03550**, 2023. Disponível em: <<https://arxiv.org/abs/2302.03550>>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016.

HAQUE, M. A. A brief analysis of “chatgpt” – a revolutionary tool designed by openai. **EAI Endorsed Transactions on AI and Robotics**, v. 1, p. e15, Mar. 2023. Disponível em: <<https://publications.eai.eu/index.php/airo/article/view/2983>>.

HE, K. et al. Deep residual learning for image recognition. **arXiv preprint arXiv:1512.03385**, 2015. Disponível em: <<https://arxiv.org/abs/1512.03385>>.

HESAMIAN, M. H. et al. Deep learning techniques for medical image segmentation: Achievements and challenges. **Journal of Digital Imaging**, v. 32, n. 4, p. 582–596, 2019.

IBM. **What is artificial intelligence in medicine?** IBM, Acessado em 23/06/2023. Disponível em: <<https://www.ibm.com/topics/artificial-intelligence-medicine#:~:text=How%20is%20artificial%20intelligence%20used,health%20outcomes%20and%20patient%20experiences>>.

INCE, T. et al. Real-time motor fault detection by 1-d convolutional neural networks. **IEEE Transactions on Industrial Electronics**, v. 63, n. 11, p. 7067–7075, 2016.

INCEPTION v2. 2023. <<https://paperswithcode.com/method/inception-v2>>. Accessed: 18-11-2023.

KALLENBERG, M. et al. Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1322–1331, 2016.

KERMANY, D. S.; ZHANG, K.; GOLDBAUM, M. **Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification**. Mendeley Data, 2018. Disponível em: <<https://dx.doi.org/10.17632/RSCBJBR9SJ.2>>.

MENZE, B. H. et al. The multimodal brain tumor image segmentation benchmark (brats). **IEEE Transactions on Medical Imaging**, v. 34, n. 10, p. 1993–2024, 2015.

NASR-ESFAHANI, E. et al. Melanoma detection by analysis of clinical images using convolutional neural network. In: **2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**. [S.l.: s.n.], 2016. p. 1373–1376.

PEREIRA, S. et al. Brain tumor segmentation using convolutional neural networks in mri images. **IEEE Transactions on Medical Imaging**, v. 35, n. 5, p. 1240–1251, 2016.

PHILLIPS, N. A. et al. Chexphoto: 10,000+ photos and transformations of chest x-rays for benchmarking deep learning robustness. In: ALSENTZER, E. et al. (Ed.). **Proceedings of the Machine Learning for Health NeurIPS Workshop**. PMLR, 2020. (Proceedings of Machine Learning Research, v. 136), p. 318–327. Disponível em: <<https://proceedings.mlr.press/v136/phillips20a.html>>.

PWC. **PwC's Global Artificial Intelligence Study: Exploiting the AI Revolution**. 2017. Disponível em: <<https://www.pwc.com/gx/en/issues/data-and-analytics/publications/artificial-intelligence-study.html>>.

RAJPURKAR, P. et al. **CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning**. 2017.

RAVÌ, D. et al. Deep learning for health informatics. **IEEE Journal of Biomedical and Health Informatics**, v. 21, n. 1, p. 4–21, 2017.

RAYAN, J. C. et al. Binomial classification of pediatric elbow fractures using a deep learning multiview approach emulating radiologist decision making. **Radiology: Artificial Intelligence**, v. 1, n. 1, p. e180015, 2019. PMID: 33937781. Disponível em: <<https://doi.org/10.1148/ryai.2019180015>>.

RIBEIRO, E.; UHL, A.; HÄFNER, M. Colonic polyp classification with convolutional neural networks. In: **2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS)**. [S.l.: s.n.], 2016. p. 253–258.

SHARMA, A. et al. Enhanced watershed segmentation algorithm-based modified resnet50 model for brain tumor detection. **BioMed Research International**, Hindawi, v. 2022, p. Article ID 7348344, 2022. Disponível em: <<https://downloads.hindawi.com/journals/bmri/2022/7348344.pdf>>.



SUN, W. et al. Enhancing deep convolutional neural network scheme for breast cancer diagnosis with unlabeled data. **Computerized Medical Imaging and Graphics**, v. 57, p. 4–9, 2017. ISSN 0895-6111. Recent Developments in Machine Learning for Medical Imaging Applications. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0895611116300696>>.

SZEGEDY, C. et al. Going deeper with convolutions. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2015. p. 1–9.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. **arXiv: Computer Vision and Pattern Recognition**, 2015. Disponível em: <<https://arxiv.org/abs/1512.00567v3>>.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2016. Disponível em: <<https://dx.doi.org/10.1109/CVPR.2016.308>>.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. **Machine Learning Refined: Foundations, Algorithms, and Applications**. 2. ed. [S.l.]: Cambridge University Press, 2016.

WERBOS, P. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, v. 78, n. 10, p. 1550–1560, 1990.

World Health Organization. **Pneumonia**. 2023. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/pneumonia#:~:text=Fran%C3%A7ais%20%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9%20Espa%C3%B1ol%20Key%20facts,and%20by%20addressing%20environmental%20factors>>.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: FLEET, D. et al. (Ed.). **Computer Vision – ECCV 2014**. Cham: Springer International Publishing, 2014. p. 818–833. ISBN 978-3-319-10590-1.